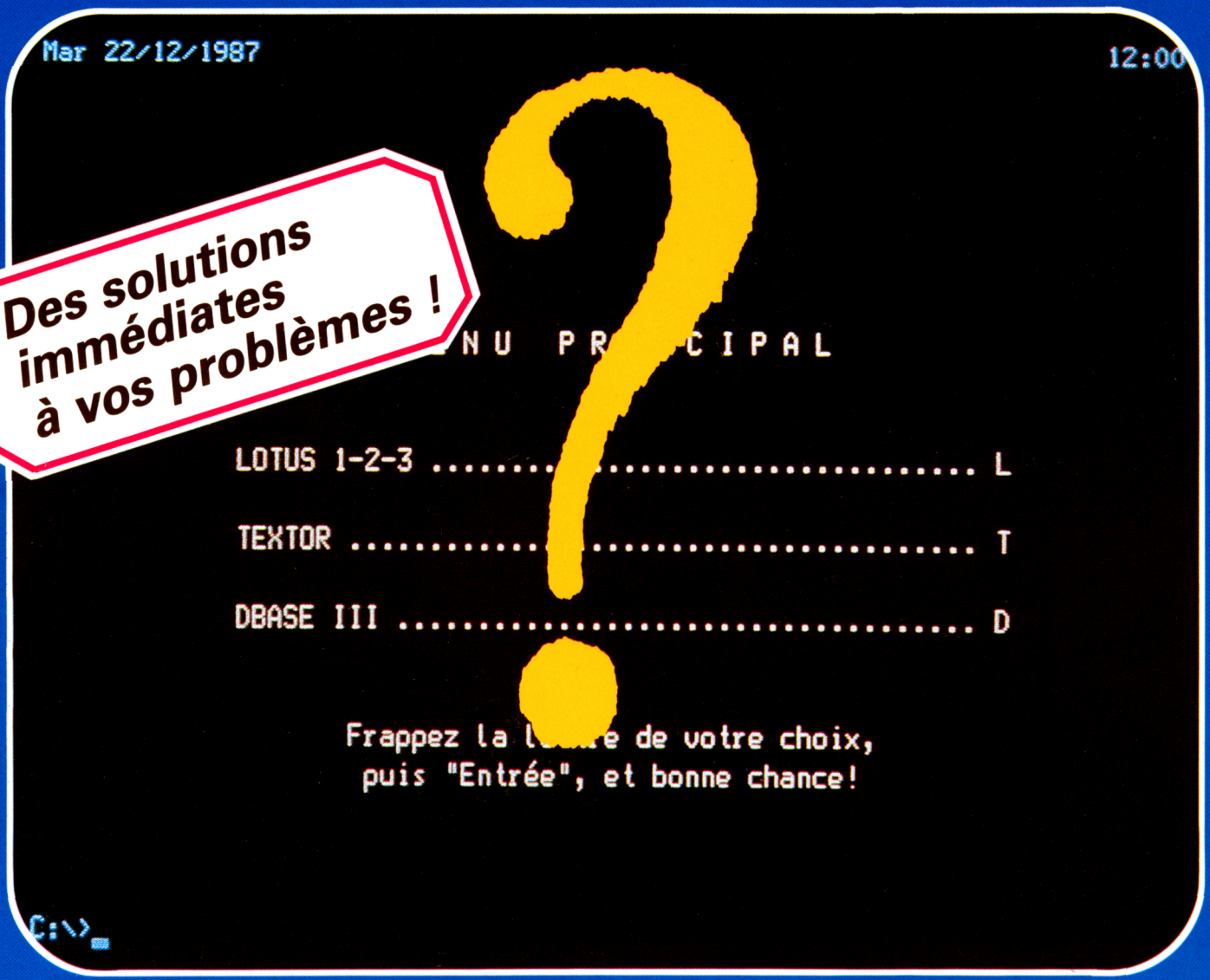
H. LILEN

50 RECETTES DOS



ÉDITIONS RADIO

ÉDITIONS RADIO

50 RECETTES DOS

H. LILEN

50 RECETTES DOS

ÉDITIONS RADIO

189, RUE SAINT-JACQUES - 75005 PARIS
TEL. (1).43.29.63.70

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1er de l'article 40). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code pénal.

Éditions Radio - Siège social : 103, boulevard Saint-Michel - 75005 Paris

© Éditions Radio, Paris
1988

Tous droits de traduction, de reproduction et d'adaptation réservés pour tous pays.

Imprimé en France par Berger-Levrault, Nancy

Dépôt légal : février 1988
Editeur nº 1094 - Imprimeur : 775195
I.S.B.N. 2 7091 1024 5

Avant-propos

Ce recueil de 50 recettes est destiné à vous faciliter le travail. Il regroupe des solutions simples et efficaces à autant de problèmes qui se présentent aux utilisateurs des systèmes d'exploitation MS-DOS ou PC-DOS sur leurs micro-ordinateurs PC/XT/AT/PS et compatibles.

Pour cela, nous avons sélectionné les questions d'ordre le plus général que se posent aussi bien les débutants que les utilisateurs expérimentés, en développant à l'aide d'exemples réels et concrets les solutions qu'il convient de leur apporter.

Les problèmes ont été classé par catégories (et par ordre alphabétique) : clavier, commandes (dont les problèmes « divers »), confort, disques, écran, fichiers, imprimante, mémoire et répertoires. Nous avons indiqué, pour chaque fiche, à qui elle s'adressait : aux débutants, aux utilisateurs déjà avertis (initiés) ou aux experts encore qu'il ne faille pas trouver là de frontières absolues.

1

Economisez les frappes des commandes répétitives grâce aux touches d'édition

Problème

On veut répéter une série de commandes identiques en salve. Par exemple, on dispose de 15 disquettes à récupérer ; on veut d'abord effacer totalement leur contenu, avec un `DEL A: *.*`. *Cela fait, on les vérifiera toutes avec un* `CHKDSK A:`. *Faut-il refrapper 15 fois ces mêmes ordres (au risque de se tromper, ce qui est très dangereux avec* `DEL` !) ?

Solution

Dans leur sagesse infinie, les concepteurs des PC ont réservé à certaines touches de fonction des rôles de *répétition.* L'auteur utilise surtout F1 et F3, et vous les recommande vivement :

F1

- F1 répète la *dernière* commande, caractère par caractère (une frappe ré-affiche un caractère).

F3

- F3 répète la *dernière* commande, en bloc, d'un seul coup.

Ainsi, et pour effacer le contenu d'une série de disquettes, frappez *une seule fois :*

```
C:\> DEL A:*.*
```

Après confirmation et exécution, changez de disquette et ne frappez plus que sur F3. Vous recommencez ensuite la série avec :

```
C:\> CHKDSK A:
```

Notez qu'avec F1, qui reprend les caractères de la dernière commande un par un, on peut en effacer en cours de route avec la touche *Del* (ou *Suppression*), ou en ajouter avec la touche *Ins*, pour *Insérer*. Cela fait, la frappe de F3 ré-affichera *le reste* de la commande, s'il y a lieu, et si vous avez bien compté vos suppressions et insertions afin de respecter le bon nombre de caractères.

Voici d'ailleurs la liste des touches utilisables en mode *d'édition*, sous DOS :

TOUCHES D'EDITION SOUS DOS

Touche	Fonction
F1 (ainsi que la touche de gestion du curseur →)	Recopie un caractère à la fois
F2 *caractère*	Recopie tous les caractères jusqu'à celui spécifié (noté ici *caractère*)
F3	Recopie tout le contenu du buffer
F4 *caractère*	Recopie tous les caractères *à partir* de celui spécifié
F5	Enregistre la ligne saisie dans le buffer mais ne l'exécute pas
F6	Frappe Ctrl + Z (^Z), caractère marquant la fin d'une séquence, d'un fichier
DEL	Efface le caractère marqué par le curseur ou supprime le caractère suivant dans le buffer
Esc	Annule la ligne affichée, mais ne modifie pas le buffer. Place un trait oblique inversé à l'écran et ramène le curseur à la ligne
Ins	Insère de nouveaux caractères

2

Comment afficher les caractères ASCII ne figurant pas sur le clavier ?

Problème

On veut tracer un cadre, ou encore utiliser des caractères grecs ne figurant pas sur le clavier, mais existant dans le code ASCII qui regroupe tous les caractères utilisables. Comment les obtenir ?

Solution

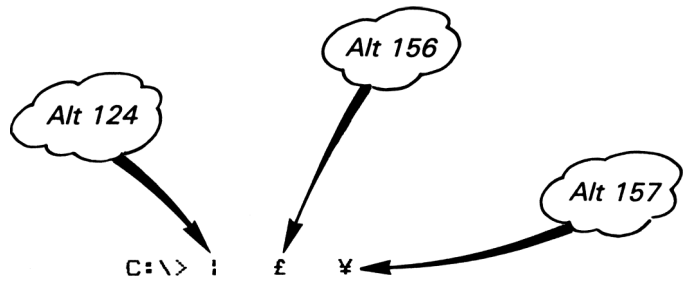
Les caractères qui apparaissent sur le clavier sont, pour l'essentiel, codés par les combinaisons ASCII allant jusqu'au 127. Le code ASCII, pour *« American Standard Code for Information Interchange »*, soit *code américain standard pour l'échange d'information*, est toutefois souvent étendu jusqu'à 225, offrant 256 combinaisons (de Ø à 255). C'est qu'en effet, les mots d'un octet, dont l'usage s'est généralisé, permettent de coder directement ces 256 combinaisons.

Malheureusement et dans le meilleur des cas, ou presque, un clavier ne comporte que 102 touches. Il reste donc encore bon nombre de codes qui ne sont pas directement accessibles. Pour les obtenir, il faut passer par une combinaison avec la touche *Alt*.

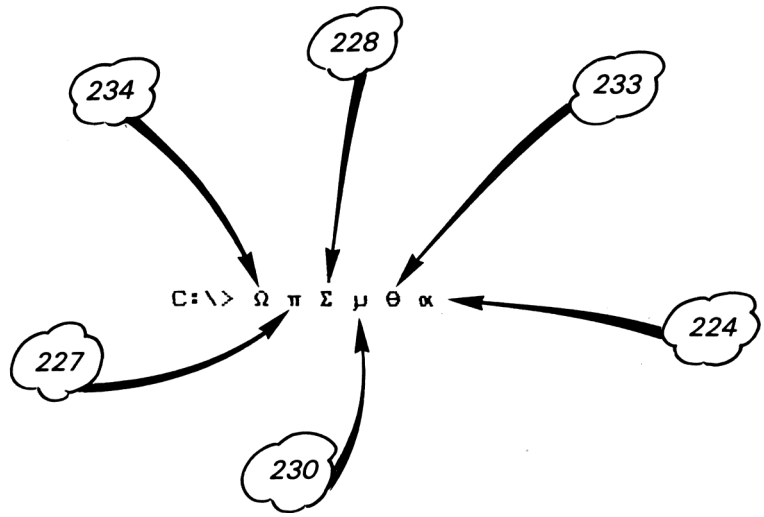
Pour obtenir un caractère avec *Alt* :
· Maintenez *Alt* enfoncée.
· Frappez le code ASCII du caractère sur le bloc numérique (de droite) du clavier.
· Relâchez *Alt* : il s'affichera à l'écran.

Attention : *n'utilisez pas* les touches numériques du clavier principal alphabétique, car cela ne marcherait pas.

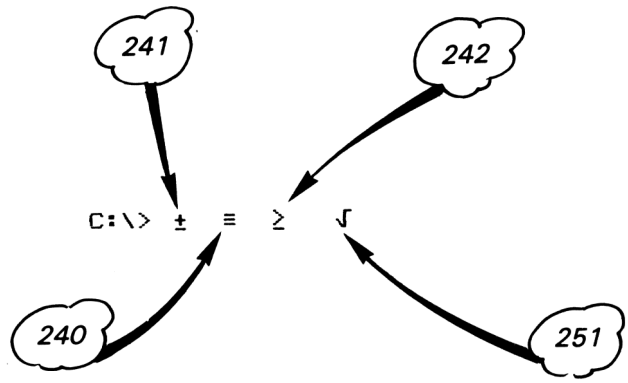
En voici quelques exemples, avec leurs codes, d'abord pour la *barre verticale coupée* (code 124) qui sert à chaîner des commandes sous DOS, puis pour les symboles monétaires *livre* (156) et *yens* (157) :



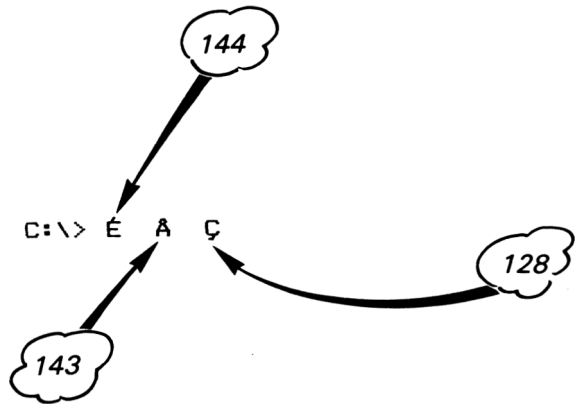
Quelques caractères grecs :



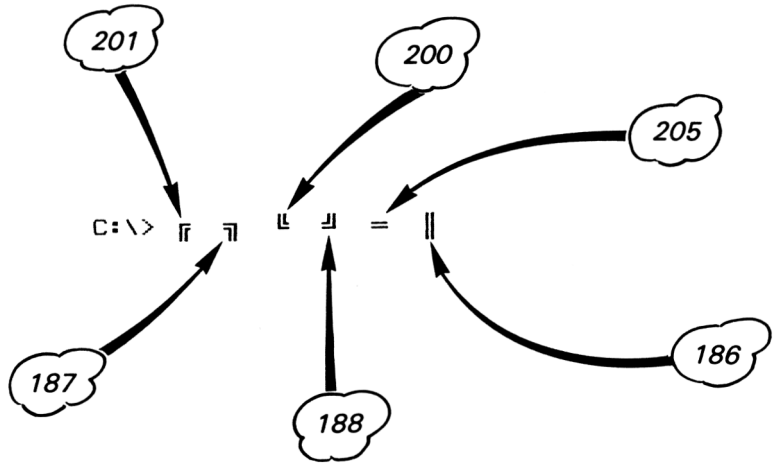
Des symboles mathématiques :



Des lettres spéciales :



Pour tracer des cadres en filets doubles :



Nous avons consacré une fiche spéciale au tracé des cadres et des filets. Pour les autres combinaisons ASCII, référez-vous à notre tableau, en fin d'ouvrage.

3

Comment tracer des cadres et filets parfaits

Problème

Sous différents programmes, tels que des traitements de texte, on aimerait tabloter, tracer des cadres, des filets verticaux parfaits, etc. Peut-on le faire avec les caractères existants (sans recourir aux points d'exclamation pour dessiner des filets verticaux !) ?

Solution

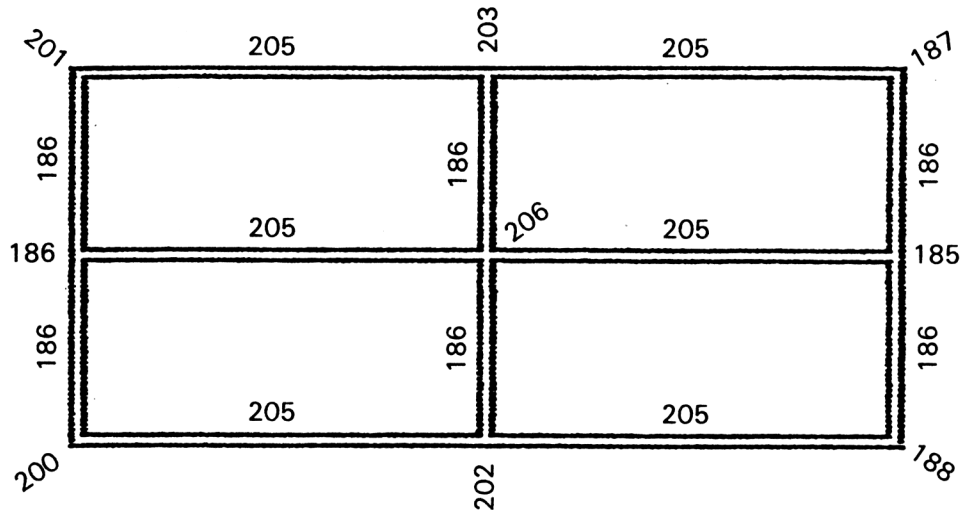
Le jeu des caractères *IBM* disponible au clavier permet effectivement de dessiner des cadres et traits, simples ou doubles, avec des angles se raccordant très exactement.

Ces caractères ne sont toutefois pas obtenus via une frappe *directe*, mais avec l'aide de la touche *Alt*. La règle est la suivante :

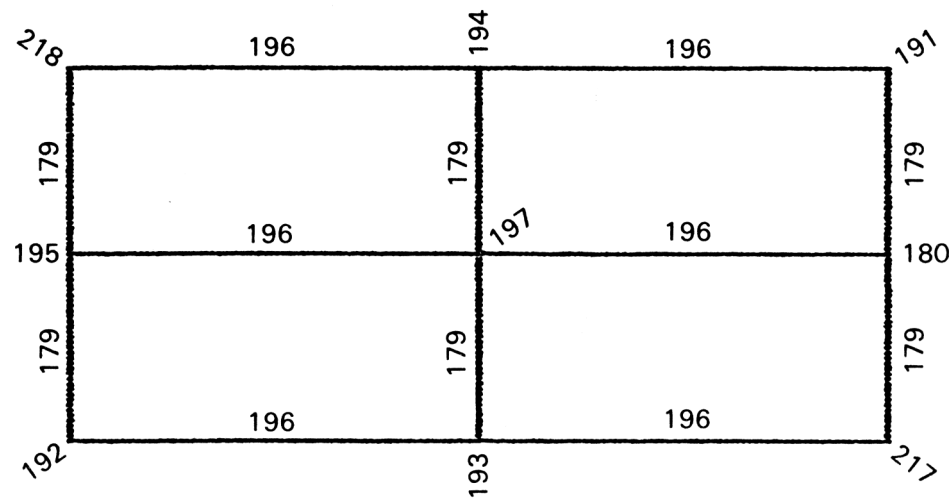
- **Maintenez Alt enfoncée.**
- **Frappez un code sur le pavé numérique de droite du clavier, exclusivement.**

Ces codes sont ceux du tableau ASCII, entre 128 et 255. Ils permettent le tracé de droites horizontales et verticales d'angles vifs et d'amorces de séparations, en traits simples et doubles. En voici quelques exemples ; les codes portés ici correspondent aux filets obtenus en regard :

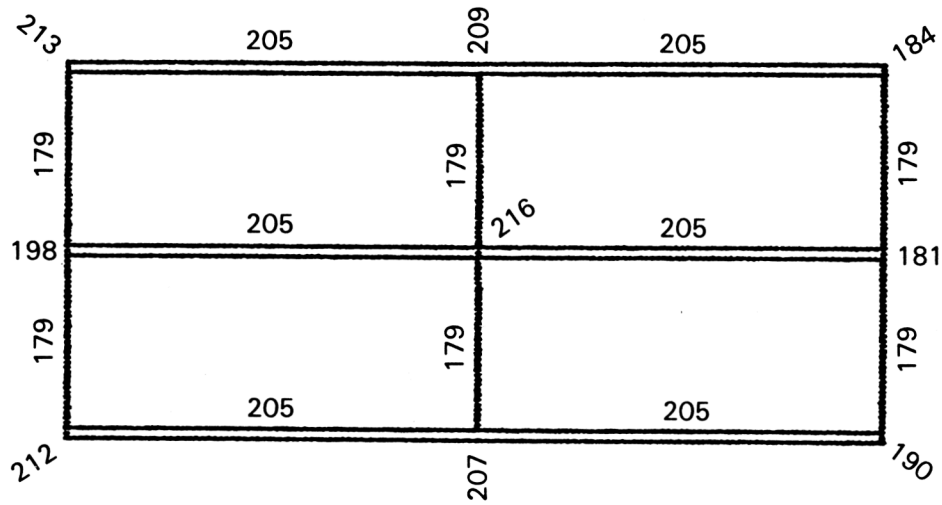
EXEMPLE 1



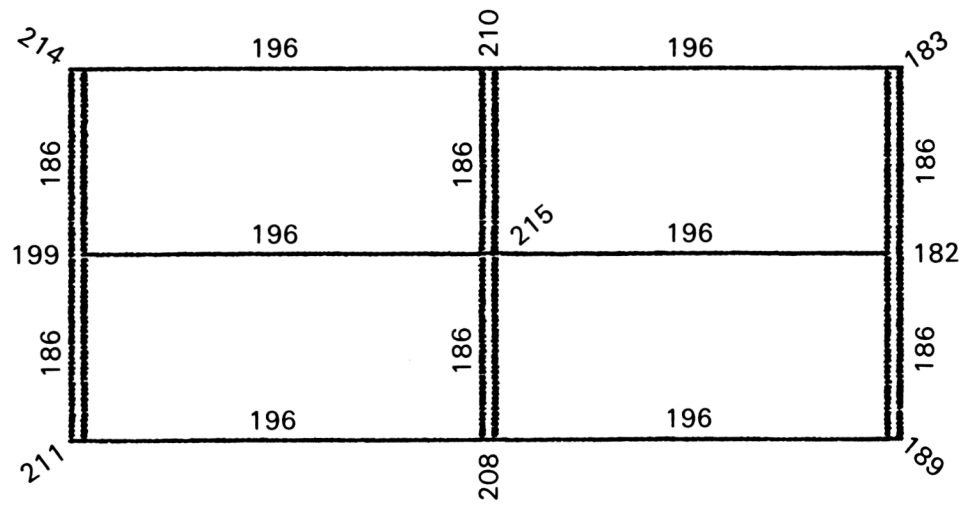
EXEMPLE 2



EXEMPLE 3



EXEMPLE 4



Méfiez-vous : ne frappez pas ces codes, avec *Alt* enfoncée sur la ligne numérique du clavier principal, car cela ne fonctionnerait pas !

Vous pouvez ainsi tracer des cadres et filets à volonté, le plus souvent sous la plupart des traitements de texte et sous bien d'autres progiciels *qui ne modifient pas les attributions des touches.*

4

Comment démarrer avec le pavé numérique du clavier 102 touches en mode numérique

Problème

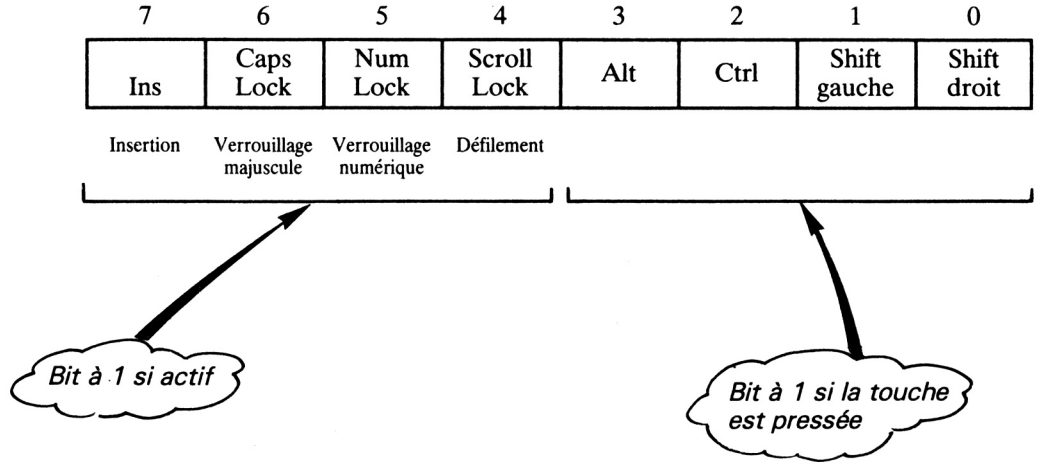
On dispose d'un clavier 102 touches, type AT3 ou PS/2. Puisqu'il existe un bloc de touches de gestion du curseur distinct, il est plus intelligent de réserver les touches numériques de droite au mode numérique. Or, à la mise en service, elles se positionnenent généralement en mode non numérique. Comment leur imposer l'état souhaité à la mise sous tension ?

Solution

Voici le moyen d'obtenir un petit confort supplémentaire. Commençons par quelques courtes explications techniques, que vous pouvez toutefois sauter si elles ne vous intéressent pas afin de passer directement à la recette à appliquer.

Tous les états du clavier son représentés par un octet géré par le BIOS. On peut l'examiner en passant par `INT 16H` , après avoir chargé 2 dans AH. Il faut se reporter au

listage du BIOS (fourni dans les manuels de références techniques du DOS) pour le localiser. Avec les PC/XT/AT, il se trouve à l'adresse **40 : 17** , en hexadécimal. Son organisation est la suivante :

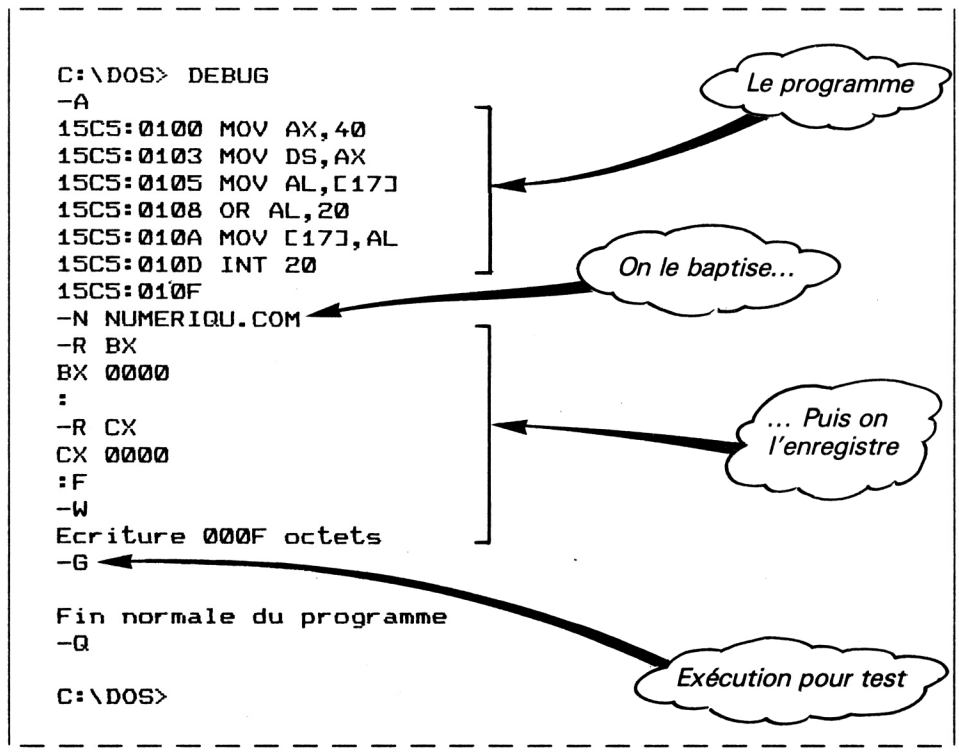


Par conséquent et pour verrouiller en position numérique, il faut écrire un 1 dans le bit 5 sans toucher aux autres bits : c'est l'affaire d'un masquage **OR**, avec le nombre hexadécimal **20H**, soit **0010 0000** en binaire.

VERROUILLER EN POSITION NUMERIQUE

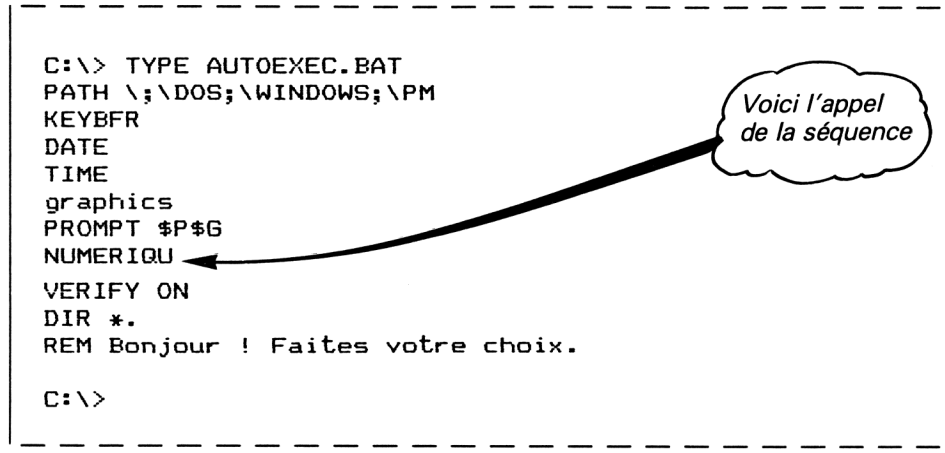
DEBUG

La séquence peut être rédigée sous *Debug*. On nomme ce programme **NUMERIQU.COM**. Même si vous n'entendez rien à *Debug*, respectez scrupuleusement les frappes suivantes, dans lesquelles les « adresses » du type « **15C5 : 0100** » vous sont imposées et écrites par *Debug* (les vôtres pourront être différentes, c'est sans importance pour les 4 premiers chiffres ; car les 4 derniers doivent obligatoirement être **0100** pour commencer la liste). N'oubliez pas les *Entrées* à la fin de chaque ligne : si vous vous trompez, corrigez aussitôt ou, en désespoir de cause, frappez une *Entrée*, puis quittez *Debug* en faisant Q puis *Entrée*, et recommencez tout :



Avant de quitter Debug, avec le Q final, vérifiez le fonctionnement de cette séquence. Les deux premières lignes chargent 4Ø dans le registre sélecteur de segment ; puis, on appelle le contenu de l'adresse 17 dans AL et cela, en adressage *direct* ; n'oubliez pas les crochets autour de l'adresse. On exécute alors un masquage avec le OR qui se borne à mettre à 1 le bit de verrouillage numérique, s'il n'y était déjà, sans modifier les autres. Puis, on renvoie le nouveau mot en 17 et on termine la séquence.

La dernière étape consiste à introduire l'appel de **NUMERIQ.COM** dans le fichier **AUTOEXEC.BAT** du répertoire principal, lequel pourrait avoir l'aspect suivant :



Si vous ne savez comment écrire **NUMERIQU** dans **l'AUTOEXEC.BAT**, reportez-vous à notre recette consacrée à ce type de fichier.

Une dernière remarque : méfiez-vous des rares compatibles où l'adresse de l'octet du mot d'état clavier pourrait être différente de celle donnée ici.

5

Comment programmer les touches de fonction et les autres avec ANSI.SYS

Problème

On n'utilise pas les caractères de certaines touches. Par contre, on a fréquemment besoin d'autres caractères. On aimerait pouvoir les affecter aux touches inutiles.

Solution

Il existe plusieurs méthodes de réaffectation des touches. Sous DOS, la plus directe consiste à faire appel aux commandes **ANSI**, disponibles avec les versions récentes des DOS. Vérifiez toutefois au préalable, que votre logiciel d'application ne les reprenne pas en main pour reprogrammer les touches à son tour !

Les normes ANSI 3.64.1979 ou ISO 6429 ont défini un certain nombre de fonctions qui ont été étendues au clavier. Pour exploiter ces fonctions, il faut charger le programme « driver » **ANSI.SYS** dans le fichier **CONFIG.SYS** en déclarant sur une ligne :

```
DEVICE = ANSI.SYS
```

Attention : le fichier **ANSI.SYS** doit alors se trouver dans le répertoire principal. Rappelez-vous qu'il faut ré-initialiser la machine, ensuite, avec *Alt* + *Ctrl* + *Del*, pour mettre en service le nouveau **CONFIG.SYS**.

Puis, on pourra reprogrammer les touches en introduisant les séquences correspondantes *dans un* **PROMPT**. Tous ces **PROMPT** pourront être regroupés dans un fichier **.BAT** pour un usage plus facile. Le processus est le suivant :

· On frappe **PROMPT**, et on le fait suivre d'une chaîne de caractères de reprogrammation, comprenant :
· Le code d'entrée, « **ESC [** » (sans les guillemets), qui s'écrit **$E[** sous DOS, puisque E est le code de **ESC**.
· On frappe l'ancien code ASCII de la touche (65 pour A, ou 97 pour *a*, etc.).

Si l'on recourt aux codes étendus, de 128 à 255, on frappera deux codes séparés par un point-virgule : d'abord un zéro (0), puis le code du caractère.

· Séparé par un point-virgule, on frappe le nouveau code qui se substituera à l'ancien, ou encore la chaîne de caractères *encadrée obligatoirement de guillemets* qui sera affectée à la touche.
· On termine cette déclaration de reprogrammation avec un *P* (minuscule ou majuscule, peu importe).

Hélas, ce n'est pas fini : le système va, en effet, interpréter correctement la commande, mais il va aussi supposer que ce **PROMPT** est destiné à supprimer les caractères d'appel (puisque c'est avec **PROMPT** qu'on les redéfinit). Par conséquent, on sera avisé de les ajouter *à la suite*, sans espace. Nous ajouterons systématiquement **PG**.

En synthèse, les *syntaxes* principales sont ainsi, selon réaffectation :

ESC [Ancien code **;** Nouveau code**P**

ESC [Ancien code **;** " Chaîne " **;** Autre nouveau code**P**

Ou toute autre combinaison. Mais le plus simple consiste à passer à quelques exemples ; commencez par expérimenter (après avoir ajouté **DEVICE = ANSI.SYS** dans le **CONFIG.SYS**) la ligne suivante, qui va faire de la frappe du A (code 65) un Q (code 81) :

```
C:\> PROMPT $e[65;81p$P$G

C:\>
```

Les 4 derniers caractères, **PG**, servent uniquement à reformer les caractères d'appel du DOS : si vous les omettez, le curseur reviendra à la ligne *sans aucun caractère d'appel* et vous ne comprendrez plus où vous en êtes ! Avec l'*Entrée*, le A est reprogrammé : si vous le frappez, vous obtenez un Q. Par contre, le *a* reste un *a*, car on ne l'a pas modifié. Voici quelques séquences types :

```
C:\> PROMPT $e[65;81p      ----->  le A devient Q

C:\> PROMPT $e[97;113p     ----->  le a devient q

C:\> PROMPT $e[81;65p      ----->  le Q devient A

C:\> PROMPT $e[113;97p     ----->  le q devient a

C:\> PROMPT $e[0;68;"Dir";13p      ----->  F10 fournit le répertoire
```

La dernière modifie la touche de fonction F10 : si vous la frappez, vous obtenez le répertoire. Le code 13 est celui du retour à la ligne. Pour expérimenter cette dernière ligne, n'oubliez pas de reprogrammer les caractères d'appel du DOS :

```
C:\> PROMPT $e[0;68;"DIR";13p$P$G

C:\>
```

Afin de revenir à la situation antérieure, ré-initialisez le micro-ordinateur.

6

Comment reprogrammer un clavier « supplémentaire »

Problème

On veut pouvoir tracer des filets ou des cadres directement, à partir du clavier, sans passer par les frappes harassantes de « Alt quelque chose », dont on ne se souvient plus. Ou, alors, on veut disposer de caractères spéciaux, en frappe directe. Sans perdre, pour autant, le clavier habituel !

Solution

Ainsi posé, ce problème ressemble à la quadrature du cercle. Pourtant, une solution simple existe. Il faut se souvenir qu'en réalité, nous disposons de *deux claviers* différents : le clavier français, obtenu avec l'utilitaire `KEYBFR.COM`, et le clavier américain d'origine.

• On sait que, lorsqu'on est en clavier français, il faut frapper Ctrl + Alt + F1 pour revenir au clavier américain.
• A l'inverse, on passe alors du clavier américain au français avec Ctrl + Alt + F2.

Faites d'abord l'essai, « à froid » : maintenez *Ctrl* + *Alt* enfoncées et frappez F1. Désormais, la touche A fournit le Q, et *a* le *q*, etc. On est passé du clavier AZERTY ou QWERTY. Revenez au clavier français : maintenez *Ctr* + *Alt* enfoncées et frappez F2.

Or, on constate que le clavier américain est inutile ; pourquoi ne pas le récupérer pour reprogrammer les touches, *dans le seul mode américain* ? C'est ce que nous vous proposons de faire mais pour cela, il faut passer par un programme capable de lire le fichier **KEYBFR.COM** , de l'afficher en hexadécimal, et permettant de le modifier. Dans l'exemple suivant, nous nous sommes servi des *Norton Utilities* .

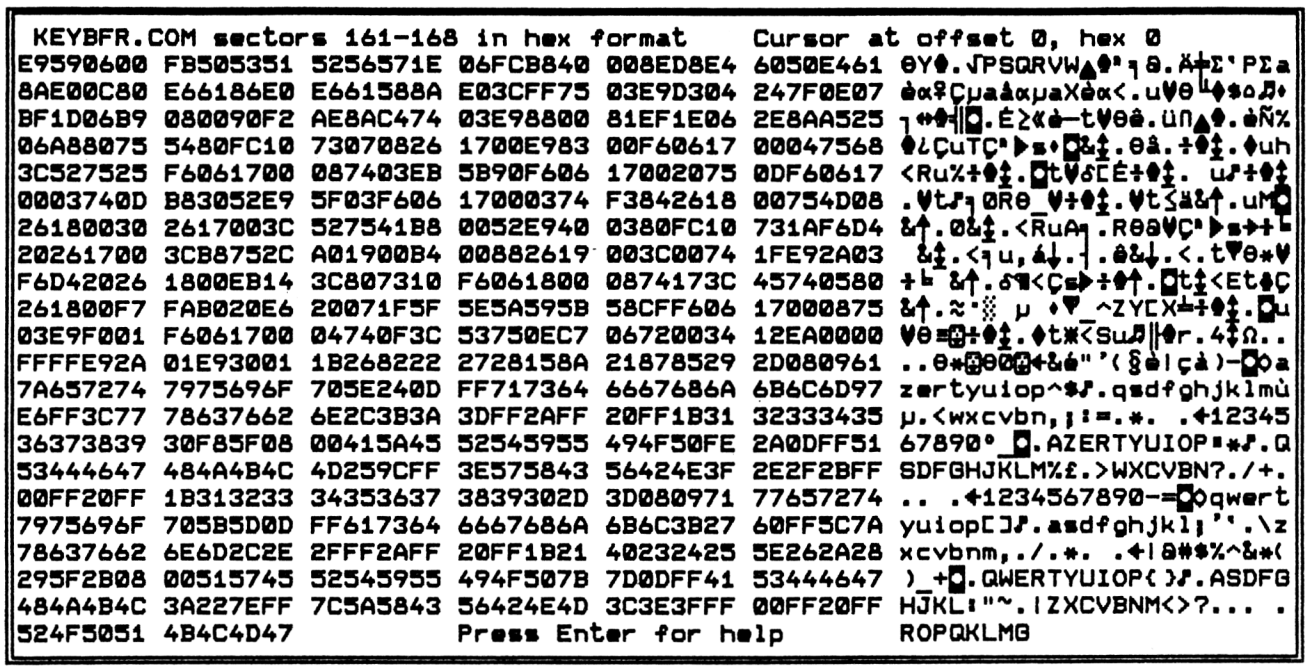
Ce que nous avons voulu faire, c'est attribuer aux touches *numériques du clavier alphanumérique*, donc à la rangée de touches supérieures, les codes permettant de tracer des filets et des cadres. Vous pouvez fort bien décider d'une autre programmation. Voici ce que l'on désire ; les lignes du centre indiquent les frappes, les lignes inférieures et supérieures les caractères obtenus :

```
╔  ═  ╦  ╗  ‖  ╟  ╬  ╢  ╚  ╩  ╝
1  2  3  4  5  6  7  8  9  Ø  °
Id. , mais avec "Shift" :
┌  ─  ┬  ┐  │  ├  ┼  ┤  └  ┴  ┘
```

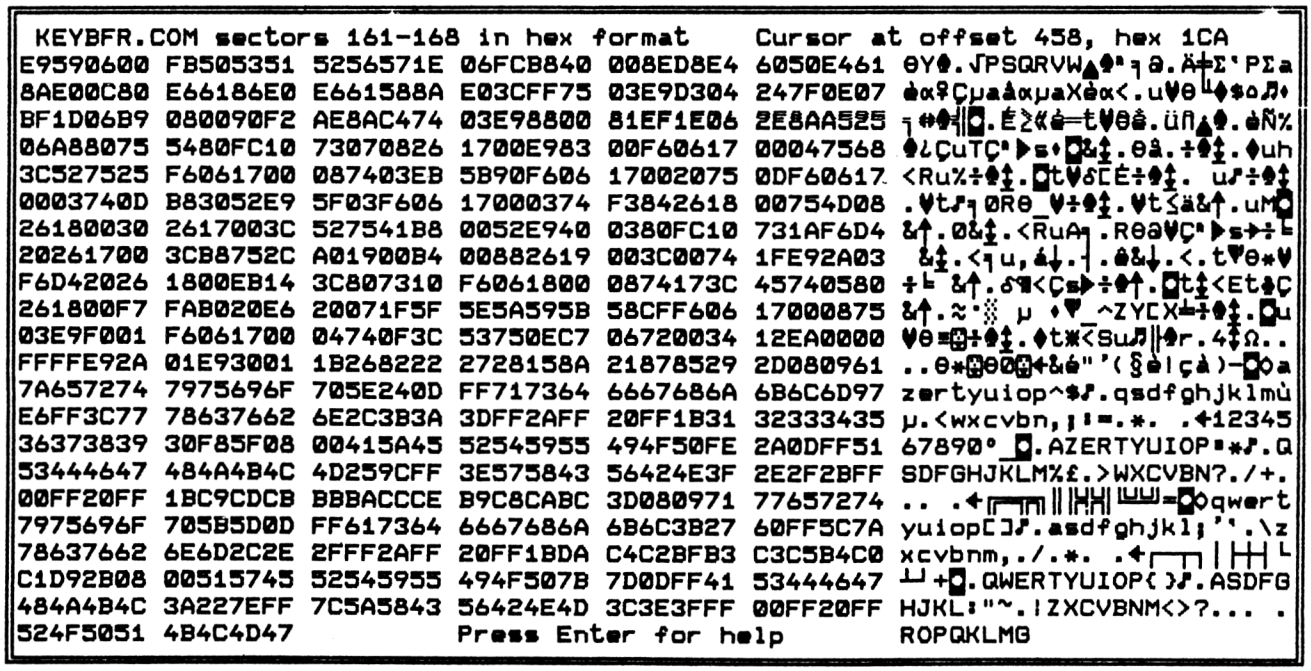
En nous référant à un tableau ASCII étendu, en hexadécimal, on a relevé les codes correspondant :

Touches	Code du caractère en hexa	
	Direct	Avec Shift
1	C9	DA
2	CD	C4
3	CB	C2
4	BB	BF
5	BA	B3
6	CC	C3
7	CE	C5
8	B9	B4
9	C8	CØ
Ø	CA	C1
°	BC	D9

La première étape consiste à faire une copie de sécurité de **KEYBFR.COM**, sous le nom de **CLAVIER** . On n'est jamais trop prudent. Puis, à l'aide du programme *Norton Utilities*, ici, on liste la page de **KEYBFR.COM** où se trouvent les codes des touches. Remarquez, à droite, les codes AZERTY, etc, et QWERTY, etc, correspondant aux claviers français *et* américain (les 8 dernières lignes) :



On va modifier les codes hexadécimaux correspondant aux touches chiffrées du clavier américain ; le symbole équivalent apparaît à droite. Ce n'est guère facile à déchiffrer mais voyez les symboles sur les lignes six, quatre et trois *à partir du bas* ; si vous comparez avec l'écran précédent, vous constaterez que les codes hexa ne sont plus les mêmes *pour ces symboles* (ce sont ceux qu'on a indiqué plus haut) !



Enregistrez les modifications, appelez **KEYBFR** et vérifiez que le clavier *français* n'a pas changé. Par contre et avec *Ctrl* + *Alt* + *F1* et donc, le clavier *américain*, les touches de la rangée numérique horizontale procurent bien ces caractères. Vous pouvez les utilisez dans quasiment tous vos logiciels, puisqu'aucun ne les modifie (d'où notre choix). Nous vous conseillons de placer une réglette au-dessus du clavier afin de marquer les nouvelles attributions.

7

Comment automatiser l'exécution de commandes avec un AUTOEXEC.BAT

Problème

Lors de chaque remise en service, ou ré-initialisation du micro-ordinateur, il faut relancer un certain nombre de commandes, toujours les mêmes. Ne peut-on automatiser cette tâche, qui devient vite fastidieuse ?

Solution

BAT.

BATCH

AUTOEXEC.BAT

L'exécution automatique d'un certain nombre de séquences ou de programmes est précisément l'objet des programmes de commande dont l'extension est `.BAT` . Ce nom provient de « *Batch* » mot qui, en anglais, désigne des *traitements par lot*, en salve.
Le principal et le tout permier des programmes `.BAT` est appelé `AUTOEXEC.BAT` , car il est à « *auto-exécution* ». Cela signifie que lors du démarrage du micro-ordinateur, le système va aussitôt le lire, puis exécuter *automatiquement* ce qu'il prescrit.
Voici l'exemple de la séquence réelle `AUTOEXEC.BAT` de l'auteur, affichée grâce à la commande `TYPE` , suivie du nom de ce qu'il faut afficher. Nous vous le commentons aussitôt après :

```
C:\> type autoexec.bat
PATH \;\DOS;\WINDOWS;\PM
KEYBFR
DATE
TIME
graphics
PROMPT $P$G
CD \DOS
BASICA ØBARRE
CD \SOURIS
MOUSE
CD \IMPRIM
HARD.COM
CD \
VERIFY ON
DIR *.
REM Bonjour ! Faites votre choix.

C:\>
```

Certaines des lignes ci-dessus font l'objet de fiches dans ce livre. Après la commande **TYPE AUTOEXEC.BAT** , on trouve ainsi :

· *Un* **PATH**, indiquant des chemins optionnels (fiche spéciale).
· *La francisation du clavier,* avec **KEYBFR.**
· *Les appels de la date et de l'heure* : oui, répondez toujours à ces demandes, car c'est facile, ça ne mange pas de pain, et ça pourra se révéler très utile en datant vos fichiers.

· **Graphics :** séquence du DOS permettant d'imprimer les caractères graphiques.

· **PROMPT PG** : l'auteur veut afficher systématiquement le nom du répertoire actif (voir fiche spéciale).

· *La ligne* **CD\DOS** fait changer de répertoire ; on passe au sous-répertoire DOS pour un fichier d'auteur, qui ordonne à son imprimante (une excellente *Star* SR 10) d'imprimer les zéros barrés (Ø). C'est le fichier **ØBARRE.**

· *Avec* **CD\SOURIS** , on passe au sous-répertoire SOURIS pour charger son fichier, appelé **MOUSE**. Nous ne traduisons pas. L'auteur emploie systématiquement une souris quand faire se peut.
· *Nouveau changement avec* **CD\IMPRIM** , pour charger le fichier **HARD.COM** servant à recopier l'écran en mode graphique sur l'imprimante.
· *Après quoi, on revient au répertoire principal,* avec **CD** .
· *Le* **VERIFY ON** *met en service une vérification automatique,* assurée par le DOS lors de la copie de fichiers. C'est un peu plus long mais c'est sécurisant (bien qu'en fait, ce soit généralement inutile !).
· *Le* **DIR *.*** *affiche les seuls noms des sous-répertoires,* listés dans le répertoire principal (voir fiche spéciale).
· Enfin, le **REM** (pour *« remarque »*) final va afficher une petite phrase gentille, destinée aux nombreux utilisateurs qui s'installent souvent devant le micro de l'auteur. Confiance oblige.

Comment créer, compléter ou modifier un tel fichier ? Nous en traitons dans diverses autres fiches : soit avec la commande **COPY** , soit sous l'éditeur **EDLIN** , par exemple.

Imaginez qu'à chaque remise en service, l'auteur doive ré-introduire toutes ces commandes au claver ; quelle perte de temps et quel souci ! N'hésitez donc pas à inclure le maximum des commandes standard dans votre **AUTOEXEC.BAT** , en vous souvenant que ces commandes ne peuvent lancer que des programmes dont le nom d'extension est **.COM** , ou **.EXE** , ou encore **.BAT** .

8

Comment passer automatiquement au programme d'application dès la mise en service du micro

Problème

On utilise le micro-ordinateur pour une comptabilité (ou une gestion de stock, ou du traitement de texte dans le cas d'un secrétariat, etc.). Comment passer directement au bon programme d'application, dès la mise sous tension du micro-ordinateur, sans aucune autre manipulation ?

Solution

AUTOEXEC.BAT

C'est précisément le rôle du fichier **AUTOEXEC.BAT** que d'automatiser (presque) tout ce qui peut l'être. Le programme **AUTOEXEC.BAT** se trouve quasi certainement dans votre DOS ; vérifiez-le avec la commande **DIR** , en frappant, par exemple :

```
C:\> DIR AUTOEXEC.BAT

 Volume dans unité C est LILEN
 Répertoire de  C:\

AUTOEXEC BAT       196  25/08/87    1:30
        1 Fichier(s)
  1277952 octets disponibles

C:\>
```

Oui, il existe !

S'il n'existait pas, ce qui est fort improbable, vous obtiendrez un message du genre :

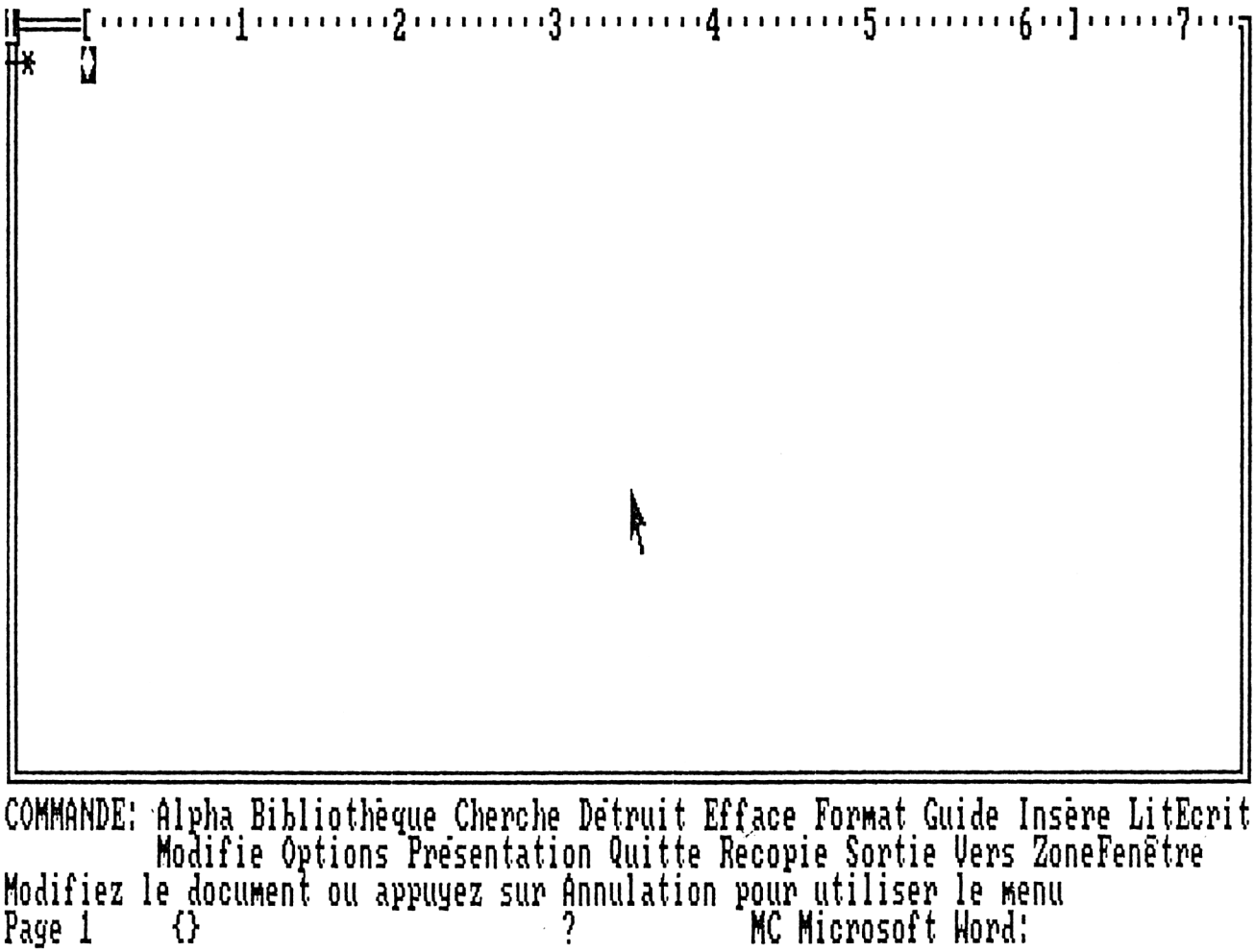
```
C:\> DIR AUTO

 Volume dans unité C est LILEN
 Répertoire de  C:\

Fichier introuvable

C:\>
```

Nous vous montrerons, dans ce cas, comment le créer. Pour l'instant, vous devez savoir que l' **AUTOEXEC.BAT** est le *tout premier* programme que va lire le micro-ordinateur, dès sa mise en service. Il y trouve des commandes qu'il exécute *aussitôt*, sans aucun délai. Par conséquent, si la dernière commande de l' **AUTOEXEC.BAT** lui ordonne de lancer votre programme d'application, il l'exécutera sans discuter.
Supposons qu'il s'agisse de lancer un traitement de texte, WORD, par exemple. L' **AUTOEXEC.BAT** pourra alors présenter cet aspect (nous le commentons aussitôt après) ; *le vôtre sera certainement différent* :

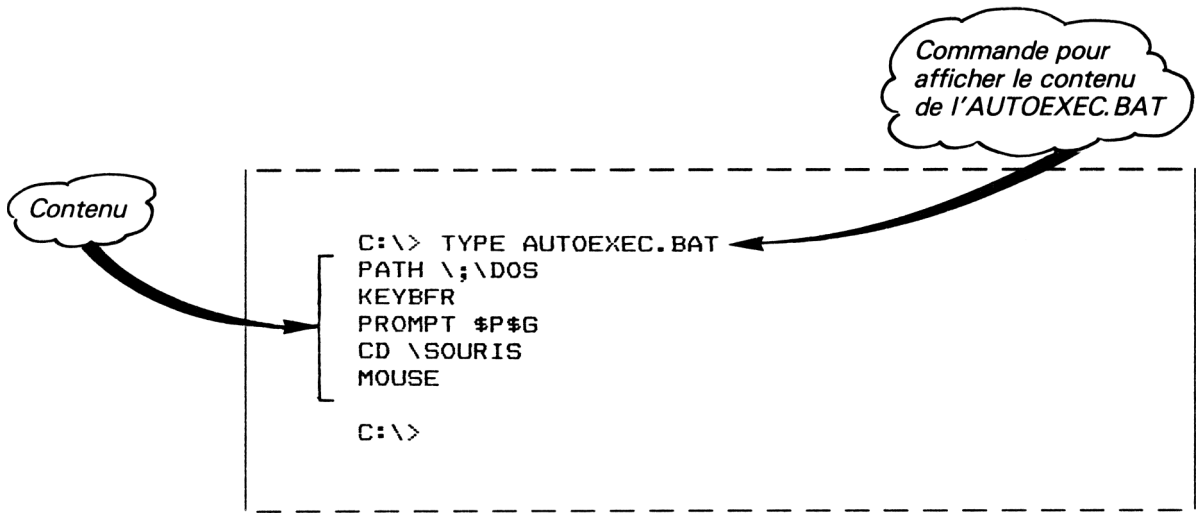
```
C:\> TYPE AUTOEXEC.BAT
PATH \;\DOS
KEYBFR
PROMPT $P$G
CD \SOURIS
MOUSE
CD \TEXTE
WORD

C:\>
```

• La ligne **Type AUTOEXEC.BAT** ordonne l'affichage de son contenu.
• **PATH \ ; \ DOS** explique au système que, s'il ne trouve pas le fichier qu'il recherche, il doit se retourner vers le répertoire principal (\ ) ou le sous-répertoire DOS (**\DOS**). Une fiche particulière traite de ces options de remplacement.
• **KEYBFR** appelle le programme francisant le clavier. Parce qu'il est incontournable, vous disposez déjà d'un **AUTOEXEC.BAT** !
• **PROMPT PG** provoque l'affichage du répertoire actif (voyez la fiche spéciale consacrée à se sujet).
• **CD \SOURIS** , puis **MOUSE** , envoie le système dans le sous-répertoire **SOURIS** , où il va charger le fichier **MOUSE** (*souris*, en anglais mais vous le saviez !). Dès lors, la souris devient opérationnelle.
• Enfin, **CD \TEXTE** renvoie au sous-répertoire de traitement de texte, alors que **WORD** lance notre progiciel.

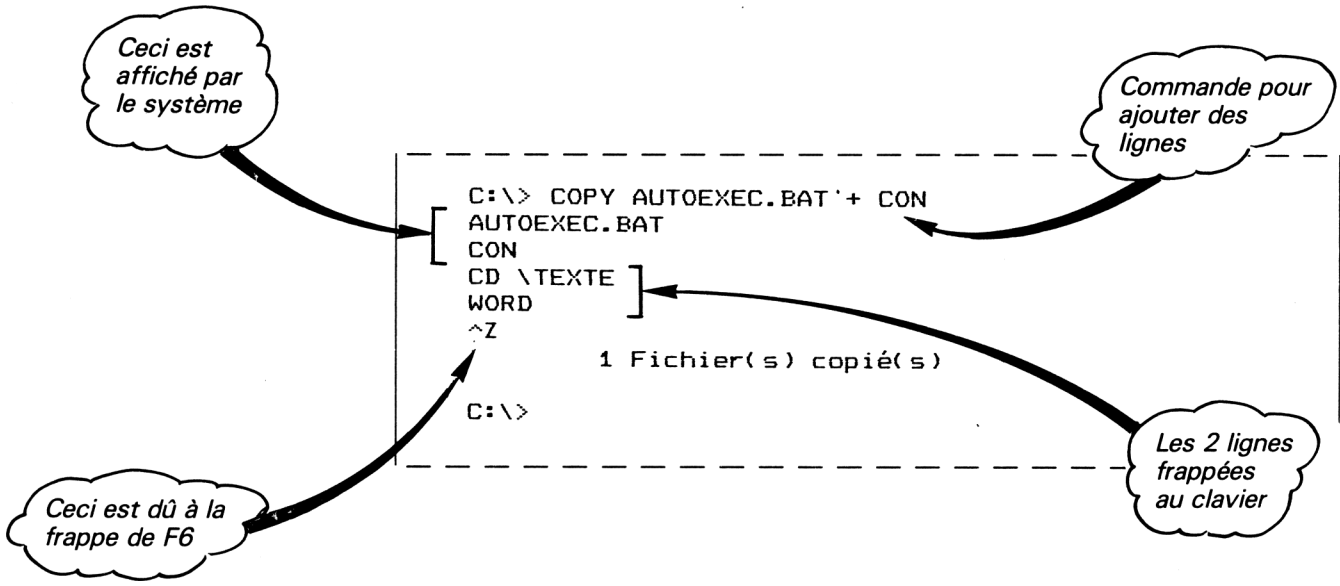
Dès la mise en service du micro-ordinateur, et après l'auto-test qu'il exécute habituellement, on verra ainsi apparaître automatiquement l'écran typique de Word :



Voyons, maintenant, comment procéder pour ajouter des commandes à l' **AUTOEXEC.BAT** . Supposons qu'au départ, il se présente ainsi :



Pour ajouter des lignes (deux, dans notre cas), on fera :



Frappez ce qui est indiqué ci-dessus, sans omettre une *Entrée* à la fin de chaque ligne ; les symboles **^Z** sont obtenus grâce à la frappe de la touche de fonction F6. Vous disposerez désormais de l' **AUTOEXEC.BAT** que nous vous avons présenté au début de cette fiche. Si vous voulez en savoir davantage sur ce type de fichier, reportez vous aux autres fiches qui l'évoquent (*Comment modifier ou créer un fichier*, *Comment automatiser des commandes*, etc.).

9

Comment lire un répertoire qui ne tient pas en un seul écran

Problème

On demande le répertoire d'un disque, avec un **DIR** *, et une liste de fichiers défile à l'écran, tellement il y en a, sans qu'on ait le temps de les lire. Comment remédier à une telle situation ?*

Solution

DIR

DIR/P

La commande **DIR** du DOS dispose de deux « clés », qui sont **/P** et **/W** :

• Si l'on frappe **DIR /P** , un écran va se remplir, puis le système marque une « pause », attend que vous frappiez sur une touche quelconque pour afficher la suite (et vous l'indique). Par exemple et avec un :

```
C:\> DIR /P
```

L'auteur a obtenu, pour son répertoire principal :

```
COMMAND  COM     25947  18/03/87   12:00
VENTE          <DIR>     1/07/87    3:43
DOS            <DIR>    17/04/87    9:22
TRANSFER       <DIR>     1/01/80    0:43
COUNTRY  SYS     11285  18/03/87   12:00
COPIE          <DIR>    17/07/87    1:12
IMPRIM         <DIR>    17/04/87    9:35
SOURIS         <DIR>    17/04/87    9:36
ADDICIEL       <DIR>    17/04/87    9:38
QR             <DIR>    17/04/87    9:39
KEYBFR   COM     1732  11/03/84   12:00
TB             <DIR>     1/01/80    0:01
WINDOWS        <DIR>     1/01/80    2:25
PCTOOLS        <DIR>    17/07/87    1:14
PAO            <DIR>    17/08/87    9:54
MP             <DIR>    22/07/87    8:54
XTREE          <DIR>     1/08/87    2:12
SALUTS            19  25/08/87    1:23
PMTUTOR  BAT      72   1/01/80    1:17
VDISK    SYS    3579  18/03/87   12:00
MSTOOLS        <DIR>    27/07/87    9:02
WORD           <DIR>    29/07/87    0:06
WORD     BAT      16  20/08/87    0:12
Appuyez sur une touche quelconque
```

Remarquez cette ligne

L'écran rempli, il se fige alors qu'en bas, une ligne invite à frapper une touche quelconque pour voir la suite ; cela fait, il est apparu :

```
CONFIG   SYS       43  25/08/87    0:33
AUTOEXEC BAT      196  25/08/87    1:30
       25 Fichier(s)
  1167360 octets disponibles

C:\>
```

S'il y avait eu de quoi remplir un second écran, le même processus aurait repris ; et ainsi de suite.

DIR/W

• Par contre, la frappe de la clé **/W** , ce W provenant de « *Width* », largeur, va fournir un affichage *en largeur*, comportant un moindre nombre d'informations, mais avec *cinq* noms de fichiers ou sous-répertoires par ligne ; c'est la commande **DIR /W** :

```
 Volume dans unité C est LILEN
 Répertoire de  C:\

COMMAND  COM    VENTE          DOS            TRANSFER       COUNTRY  SYS
COPIE           IMPRIM         SOURIS         ADDICIEL       QR
KEYBFR   COM    TB             WINDOWS        PCTOOLS        PAO
MP              XTREE          SALUTS         PMTUTOR  BAT   VDISK    SYS
MSTOOLS         WORD           WORD     BAT   CONFIG   SYS   AUTOEXEC BAT
       25 Fichier(s)
  1167360 octets disponibles

C:\>
```

Ce qui permet d'obtenir une vue de synthèse.

10

Comment afficher la date et l'heure

Problème

L'ordinateur conservant la date et l'heure, peut-on ré-afficher celles-ci systématiquement avec les caractères d'appel ?

Solution

PROMPT

La définition des caractères d'appel du DOS se fait avec la commande **PROMPT** . Lorsqu'on veut afficher le sous-répertoire actif, la commande est réduite à :

```
C:\> PROMPT $P$G
```

DATE

Veut-on ajouter la date, il suffit d'écrire, par exemple (avec des symboles « dollars » encadrant un **D** , pour *date*) :

```
C:\>PROMPT $D$ - $P$G
Mar  1/09/1987- C:\>
```

Le résultat

On peut enjoliver et spécifier le mot **Date** avec :

```
C:\> PROMPT Date = $D$ - $P$G
Date = Mar  1/09/1987- C:\>
```

Résultat

HEURE

Si l'on ajoute l'heure, avec des retours à la ligne (le petit tiret qui apparaît est celui de *soulignement*), on aboutit, dans cet exemple, à :

```
C:\> PROMPT Date = $d$_Heure = $T$_$P$G

Date = Mar  1/09/1987
Heure = 18:48:32,14
C:\>
```

Indiquée à la centième de seconde, l'heure offre une précision inutile ! Vous pouvez modifier les messages à volonté, sur ces mêmes principes, le tableau suivant fournissant la liste des codes immédiatement reconnus par `PROMPT`

CARACTERES POUR PROMPT

Caractère	Fonction
$	Le caractère « dollars » qui doit obligatoirement précéder les autres
T	L'heure (« Time »)
D	La date
P	Le répertoire courant de l'unité par défaut
V	La version du DOS
N	La lettre de l'unité à disque par défaut
G	Le caractère « supérieur à » (>)
L	Le caractère « inférieur à » (<)
B	Le caractère « barre verticale coupée »
Q	Le caractère « égal » (=)
E	Le code « Escape » (1B en hexadécimal et en ASCII)
H	Un espace arrière : le caractère précédent est effacé
_	Le tiret de soulignement : il représente le couple « passage à la ligne, retour en début de ligne », souvent noté CR LF

11

Comment créer un menu pour rendre le micro encore plus convivial

Problème

A la mise en service du micro, on peut s'orienter vers diverses applications : traitement de texte, gestion de fichiers, tableur, etc. Au lieu de taper au clavier des commandes pas toujours évidentes au profane, il vaudrait mieux passer par un menu à choix multiples.

Solution

MENUS

Avec un petit peu d'imagination, le DOS offre un moyen simple de réaliser des menus. Reprenons l'exemple du choix entre un tableur, Lotus 1, 2, 3 par exemple, le traitement de texte Textor et la gestion de fichiers dBase III. On veut que l'écran affiche ce menu :

```
          M E N U   P R I N C I P A L

LOTUS 1-2-3 ..................................... L

TEXTOR .......................................... T

DBASE III ....................................... D

      Frappez la lettre de votre choix,
       puis "Entrée", et bonne chance!

C:\>
```

MENU.BAT

Puis, que la frappe de L, T ou D lance automatiquement le programme choisi, et enfin, qu'à la sortie de ce dernier, on revienne d'office au menu. Le processus va consister :

· A créer un fichier de menu, sous le nom de **MENU.BAT**, car il s'agit d'une séquence **.BAT** .
· A créer autant de fichiers **.BAT** spécialisés qu'il y a de programmes au menu.
· A introduire l'appel de **MENU.BAT** en fin de l' **AUTOEXEC.BAT** .

Pour créer **MENU.BAT**, on peut passer par un **COPY CON** de la manière suivante :

```
C:\> COPY CON MENU.BAT
ECHO OFF
CLS
ECHO            M E N U    P R I N C I P A L
ECHO.
ECHO.
ECHO LOTUS 1-2-3 ..................................... L
ECHO.
ECHO TEXTOR .......................................... T
ECHO.
ECHO DBASE III ....................................... D
ECHO.
ECHO.
ECHO        Frappez la lettre de votre choix,
ECHO         puis "Entrée", et bonne chance!
ECHO.
ECHO.
^Z
         1 Fichier(s) copié(s)

C:\>
```

ECHO

ECHO OFF

CONTROLE Z

La commande **ECHO OFF** supprime l'affichage des commandes en écho, et **CLS** efface l'écran. Pour créer des lignes vides, on a écrit **ECHO.** suivi d'un point. Le *Contrôle Z* final (**^Z**) est obtenu par la frappe de la touche de fonction F6. Un **ECHO** tout seul provoquera l'affichage de ce qui est écrit à la suite.
La seconde étape va consister à réaliser *trois* autres fichiers **.BAT**, l'un pour Lotus, le second pour Textor et le troisième, pour dBase. Leur contenu *exact* dépend de votre système ; si tout est organisé en sous-répertoires sur le disque dur, on pourra faire, pour dBase, par exemple :

```
C:\> COPY CON D.BAT
CD \DBASE
DBASE
CD \
MENU
^Z
         1 Fichier(s) copié(s)

C:\>
```

La frappe de D déclenche le passage au sous-répertoire dBase (**CD \DBASE**), puis l'appel du programme. A la sortie de dBase III, on revient au répertoire principal (**CD **), puis au menu.

Si le programme est protégé et exige la présence de la disquette d'origine, on pourra faire, par exemple pour Lotus :

```
C:\> COPY CON L.BAT
ECHO OFF
CLS
ECHO Placez la disquette Lotus en A:
PAUSE
CD \LOTUS
123
CD \
MENU
^Z
        1 Fichier(s) copié(s)

C:\>
```

La commande **PAUSE** marque un arrêt et affiche un message, ce qui procurera, ici :

```
Placez la disquette Lotus en A:
Appuyez sur une touche quelconque
```

PAUSE

La frappe d'une touche quelconque permettra de poursuivre.

Enfin, et en troisième étape, on introduira une ligne ne comportant que **MENU** à la fin de l' **AUTOEXEC.BAT**, afin que celui-ci apparaîsse à chaque ré-initialisation.

12

Substituez une simple lettre à vos longs chemins d'accès

Problème

La structure hiérarchique des fichiers sur le disque dur fait qu'on doit souvent appeler des fichiers dans le sous-sous-sous-(etc.)-répertoire :

```
C:\ALPHA\BETA\DELTA\EPSILON\OMEGA> CD \
```

La frappe de ce chemin complet est astreignante. Comment le remplacer par quelque chose de plus court ?

Solution

La commande **SUBST**, du DOS, sert à *substituer* à un long chemin, tel que celui-ci, un nom d'unité à disque fictive. Ici, si l'on adopte une unité **E :** inexistante physiquement, on écrira :

```
C:\>SUBST E: C:\ALPHA\BETA\DELTA\EPSILON\OMEGA

C:\>
```

Dès lors, si vous demandez le **DIR** de **E :** , vous obtiendrez celui d'*Oméga* (il ne contient qu'un seul fichier) :

```
C:\> DIR E:

 Volume dans unité E est LILEN
 Répertoire de  E:\

.            <DIR>      31/03/87   1:09
..           <DIR>      31/08/87   1:09
ON_Y_EST             6  31/08/87   1:10
       3 Fichier(s)
  1089536 octets disponibles

C:\>
```

C'est exactement le même que celui d'*Oméga*, demandé sous *Oméga* :

```
C:\ALPHA\BETA\DELTA\EPSILON\OMEGA> DIR

 Volume dans unité C est LILEN
 Répertoire de  C:\ALPHA\BETA\DELTA\EPSILON\OMEGA

.            <DIR>      31/08/87   1:09
..           <DIR>      31/08/87   1:09
ON_Y_EST             6  31/08/87   1:10
       3 Fichier(s)
  1089536 octets disponibles

C:\ALPHA\BETA\DELTA\EPSILON\OMEGA>
```

Si, en cours de route, vous avez oublié de quelle substitution il s'agit, demandez-le au DOS en tapant simplement **SUBST** :

```
C:\> SUBST
E: => C:\ALPHA\BETA\DELTA\EPSILON\OMEGA

C:\>
```

On peut annuler une substitution avec la clé **/D** , ainsi :

```
C:\> SUBST E: /D

C:\>
```

Dès lors, le DOS n'a plus rien à répondre à une demande d'affichage des **SUBST** en cours :

```
C:\>SUBST

C:\>
```

Comment appliquer ce principe, en pratique ? Introduisez le **SUBST** dans votre **AUTOEXEC.BAT** et ce, *avant* le **PATH** ; dans ce dernier, reprenez le nom de l'unité à disque fictive **E :**

Une ligne PATH est limitée à 127 caractères, le mot PATH compris. Ce principe permettra de loger davantage de chemins. De même, un CD (« *Change Directory* ») est limitée à 64 caractères.

PATH

En fait, le DOS réserve 160 octets à PATH et à PROMPT.

LASTDRIVE

L'économie de frappes qui résulte de l'usage de SUBST réduit les erreurs, et les énervements inutiles. Nous avons adopté E : comme unité fictive, mais vous pourriez mettre une autre lettre *à la condition d'avoir indiqué au DOS que vous utiliserez davantage d'unités à disque,* avec un **LASTDRIVE** inclus dans le **CONFIG.SYS** .

Accessoirement, rappelez-vous qu'à l'évidence, certaines commandes ne fonctionnent pas avec l'unité à disque de substitution, par exemple **LABEL** , ou **BACKUP** . Méfiez-vous en.

13

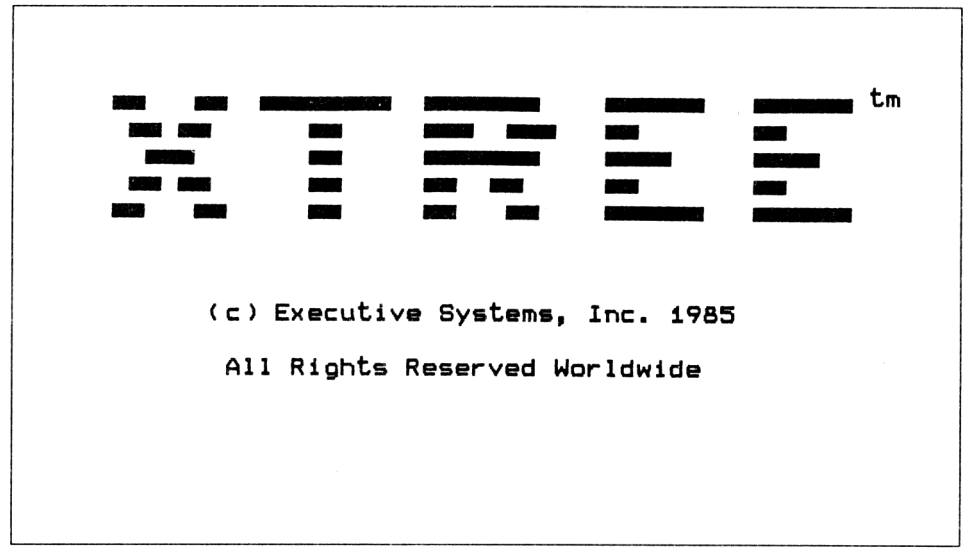
Si l'on se perd dans les multiples sous-répertoires, comment en afficher la structure ?

Problème

On a créé tellement de répertoires et sous-répertoires, sur le disque dur, qu'on s'y perd. Un bon dessin, sur l'écran, montrant toute cette organisation, serait le bienvenu. Peut-on le créer ?

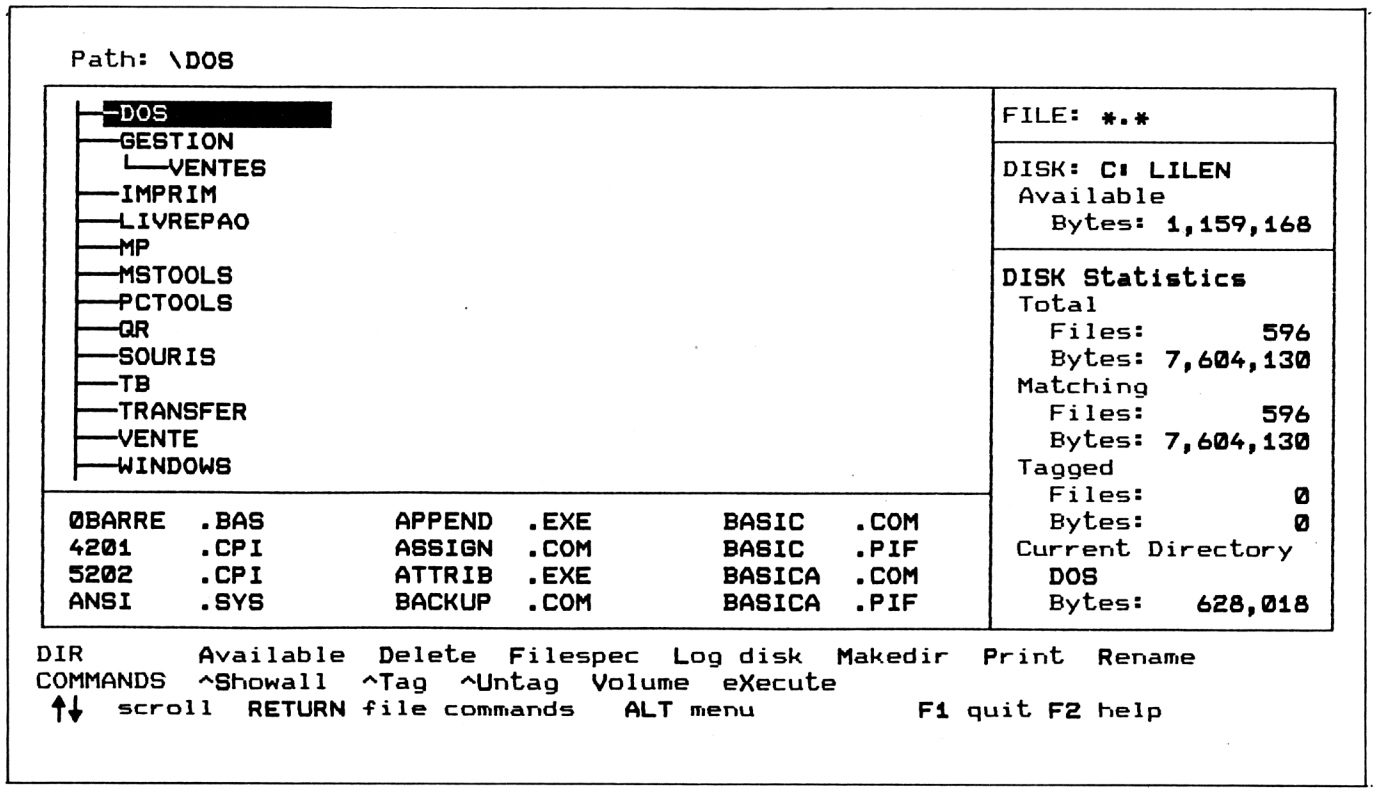
Solution

Oui, bien sûr (sinon, nous n'évoquerions pas ce problème !). Le DOS fournit bien une commande, servant à dresser le tableau de l'arborescence des répertoires, avec `TREE` (« arbre »), mais elle est peu efficace. Nous préférons plutôt vous conseiller d'acquérir un programme « utilitaire » distinct, tel qu'il en existe plusieurs, et par exemple, XTREE, d'*Executive Systems.* A son lancement, il fait apparaître un écran de présentation, dont voici l'extrait principal :



Cet écran est aussitôt suivi d'un second exposant l'organisation des sous-répertoires et sous-sous(etc)-répertoires. Dans l'illustration suivante, DOS, GESTION, etc., sont des sous-répertoires, alors que VENTES (3e ligne) est un sous-sous-répertoire, issu de GESTION.

En outre, le cadre du bas dresse la liste des fichiers contenus dans le répertoire marqué en vidéo inversée, la barre noire, ici, avec son texte blanc. On peut « entrer » dans cette fenêtre et faire défiler ces noms de fichiers, s'ils sont plus nombreux :



Ce même programme analyse (à droite) le contenu du disque spécifié. Celui de l'auteur comporte 596 fichiers. On peut afficher leur conteu, changer leurs noms, en supprimer en série, etc. ; on peut également modifier les noms des sous-répertoires, marquer ou démarquer les fichiers en agissant sur leurs « attributs », etc. Aussi, ce type de programme se révélera-t-il fort précieux à l'usage.

14

Comment permettre des réponses dans un fichier .BAT

Problème

Les fichiers « batch » (.BAT) sont très pratiques, mais ne s'ouvrent pas au dialogue. On ne peut répondre à d'éventuelles questions car ils ne savent pas tester la réponse. Le DOS n'a pas prévu cette possibilité. Pourtant, il est très facile de pallier cette insuffisance.

Solution

Elle consiste à faire appel à l'interruption INT 16 du DOS, via la fonction Ø (code à loger dans le registre AH) qui sert à la saisie d'un caractère, sans ré-affichage (sans « écho »). Pour cela, il faut développer un tout petit utilitaire, sous Debug, par exemple (c'est le plus rapide !), qu'on appellera `QUESTION.COM` . Dès lors, si le fichier `.BAT` pose une question et qu'on répond par une frappe au clavier, on pourra lire celle-ci dans le registre AL, sous forme de son code ASCII.

Mais comment tester, ensuite, AL ? C'est extrêmement simple : l'instruction `ERRORLEVEL` , qu'on peut inclure dans un fichier `.BAT` , sert précisément à cela. Vous savez désormais tout. Voici la séquence à entrer sous Debug ; il faut appeler l'*assemblage* (A) à partir de l'adresse obligatoire `100` ; celle-ci est automatiquement incrémentée, ne vous vous souciez donc plus des 8 chiffres qui démarrent chaque ligne (les 4 premiers seront certainement différents chez vous, ce qui n'a aucune importance). Les deux lignes `MOV AH, 4C` et `INT 21` interviennent pour terminer la séquence d'assemblage. Enfin, les dernières commandes `R CX` et `N` servent à enregistrer le fichier :

```
C:\> DEBUG
-A 100
1776:0100 MOV AH,00
1776:0102 INT 16
1776:0104 MOV AH,4C
1776:0106 INT 21
1776:0108
-R CX
CX 0000
:8
-N QUESTION.COM
-W
Ecriture 0008 octets
-Q

C:\>
```

Dès lors, il ne reste plus qu'à créer le fichier **.BAT** , que nous avons appelé **MENU.BAT** . Ses premières lignes affichent le menu, avec trois choix, ici ; les lignes précédées d'un double-point sont des étiquettes, pour les renvois **GOTO** . La ligne **QUESTION** lance la séquence **QUESTION.COM** créée ci-dessus : elle va lire notre frappe ; en effet, la frappe d'un 1 place le code 49 dans le registre AL, un 2 le code 50, etc. La suite du fichier **.BAT** renvoie à la séquence demandée : ici, nous n'avons fait qu'afficher une ligne mais en pratique, vous lancerez certainement un programme. On peut multiplier les options et non les limiter à 3, toutes celles non prévues correspondant à une frappe erronée ; elle est détectée par le code 52, ou au-delà, dans notre cas. La séquence **.BAT** est la suivante :

```
C:\> TYPE MENU.BAT
ECHO OFF
CLS
ECHO ------------------------------
ECHO           M E N U
ECHO ------------------------------
ECHO.
ECHO Séquence pour programme 1 =  1
ECHO Séquence pour programme 2 =  2
ECHO Séquence pour terminer    =  3
ECHO ------------------------------
:REPRISE
ECHO Frappez votre choix
QUESTION
IF ERRORLEVEL 52 GOTO NON
IF ERRORLEVEL 51 GOTO FIN
IF ERRORLEVEL 50 GOTO DEUX
IF ERRORLEVEL 49 GOTO UN
:UN
ECHO Lancement du programme 1 demandé
GOTO FIN
:DEUX
ECHO Lancement du programme 2 demandé
GOTO FIN
:NON
ECHO Répondez correctement, S.V.P. !
GOTO REPRISE
:FIN
ECHO Au revoir !
```

Le menu qui apparaît à l'écran est le suivant :

```
-----------------------------
         M E N U
-----------------------------

Séquence pour programme 1 =  1
Séquence pour programme 2 =  2
Séquence pour terminer    =  3
-----------------------------
Frappez votre choix
```

Si, par exemple, on frappe un 1, on obtiendra, dans notre cas :

```
-----------------------------
         M E N U
-----------------------------

Séquence pour programme 1 =  1
Séquence pour programme 2 =  2
Séquence pour terminer    =  3
-----------------------------
Frappez votre choix
Lancement du programme 1 demandé
Au revoir !
C:\>
```

Un 7 procurerait :

```
-----------------------------
         M E N U
-----------------------------

Séquence pour programme 1 =  1
Séquence pour programme 2 =  2
Séquence pour terminer    =  3
-----------------------------
Frappez votre choix
Répondez correctement, S.V.P. !
Frappez votre choix
```

De même pour les autres choix ; notez toutefois que la frappe d'une touche de code ASCII inférieur à 49 déclenchera l'option 1 (la barre d'espace, par exemple), et que celle d'une autre touche supérieure à 42 dans notre cas correspondra à l'option NON. Placez les conditions IF dans l'ordre inverse des codes ASCII, ainsi que nous l'avons fait, pour toute sécurité.

15

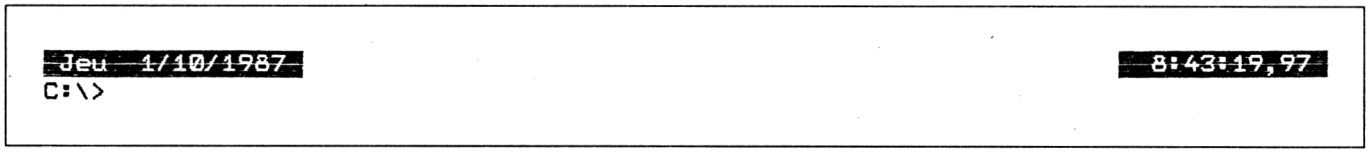
Comment afficher la date et l'heure en haut de l'écran

Problème

On veut afficher systématiquement la date et l'heure en haut de l'écran, et non plus avec les caractères d'appel du DOS ainsi qu'une autre fiche le montrait.

Solution

La solution consiste à nouveau à faire appel à la commande **PROMPT** , gérant les caractères d'appel, mais en recourant, cette fois, au code ANSI que nous avons déjà évoqué. La date et l'heure apparaîtront en haut de l'écran ainsi :



La remise à jour se fera à chaque ré-affichage des caractères d'appel.

Nous vous donnons, en annexe, la liste des codes ANSI. Vérifiez que vous disposez bien du fichier **ANSI.SYS** dans votre DOS, puis mettez-le en service en frappant dans le **CONFIG.SYS** (*et exclusivement là*) la ligne suivante :

```
DEVICE = ANSI.SYS
```

Remarquons, en passant, que l' **ANSI.SYS** est très mal documenté dans les manuels d'*IBM* ou d'autres ; aussi vous conseillons-nous de bien respecter notre façon d'écrire faute de quoi vous risquez de procédez par approximations successives (affichages curieux, corrections, etc.). Enfin, il convient de se souvenir que l'appel des codes ANSI se fait via la commande *Esc* et que celle-ci se traduit, sous DOS, par le code « **$e** » (sans les guillemets) ; ces codes commencent par une ouverture de crochet.

Afin que l'ensemble de la commande soit plus claire, nous allons la détailler, en commençant par afficher uniquement l'heure, en haut et à droite de l'écran. La commande s'écrit :

· D'abord, l'ordre **PROMPT**
· Puis, et pour définir les caractères d'appel, on place un **PG** de façon classique (**$P** indique le chemin et le répertoire actif, et **$G** place le symbole « **>** »). Il ne s'agit pas, là, des codes ANSI.
· A la suite de quoi *il faut sauvegarder* la position que le curseur a prise tout juste après les caractères d'appel, afin de pouvoir y revenir. Cette fois, on entre dans le code ANSI, et cela s'écrit **$e[s** . Rappelons que **$e** est le code *Esc* et que **[** ouvre la commande, **s** étant la *sauvegarde.*
· On positionne le curseur sur la ligne 1; colonne 68 avec un **$e[1;68H** .
· Un petit raffinement : on va passer en vidéo inversée avec un **$e[7m** .
· Puis, on affiche l'heure (en codes habituels, et non ANSI), avec un **$t** (*t = Time*).
· On annule la vidéo inversée pour revenir en affichage blanc sur fond noir avec **$e[0m** .
· Enfin, on ramène le curseur à sa position sauvegardée avec un **$e[u** .

Dès l'*Entrée*, l'heure s'affiche en haut et à droite de l'écran ; cela donne, avec la ligne complète de commande :

```
                                                              8:33:18,20
C:\> PROMPT $p$g $e[s $e[1;68H$e[7m $t $e[0m$e[u

C:\>
```

Cela compris, on va maintenant passer à l'affichage de la *date* (**$d**) et de l'*heure* (**$t**) en mettant, de surcroît, en surbrillance les caractères d'appel **C:\>** avec une double commande ANSI :

· Un **$e[1m** pour passer en surbrillance.
· Un **$e[0m** pour revenir à la brillance normale.

Le principe est rigoureusement le même que ci-dessus ; le positionnement du curseur pour la date se fait ligne 1, colonne 1. La commande étant trop longue, on a provoqué un retour à la ligne ; elle s'écrit :

```
C:\> PROMPT $e[1m$p$g$e[0m $e[s $e[1;1H$e[7m $d $e[0m $e[1;68H$e[7m $t
$e[0m$e[u
```

Dès l'*Entrée*, la date et l'heure s'affichent, sur la première ligne, et en outre, les caractères d'appel passent en surbrillance :

```
Jeu 1/10/1987                                                 8:37:25,92
C:\> PROMPT $e[1m$p$g$e[0m $e[s $e[1;1H$e[7m $d $e[0m $e[1;68H$e[7m $t
$e[0m$e[u

C:\>
```

Le code ANSI étant en service, méfiez-vous toutefois de certaines recopies d'écran, avec *Print Screen* (*Shift* + *PrtSc*) : certains caractères pourraient être convertis à tort par ce code.

16

Pour vérifier automatiquement les copies sur disque

Problème

Le fichier qu'on vient de copier d'un disque sur l'autre a-t-il été correctement enregistré ? Peut-on le vérifier sans aucune manœuvre supplémentaire ?

Solution

COMP

DISKCOMP

VERIFY

VERIFY ON

Pour vérifier des copies de fichier d'un disque sur l'autre, on dispose de plusieurs commandes, dont **COMP** , *comparer*, qui peut suivre des **COPY** , et **DISKCOMP** , qui compare des disquettes copiées avec **DISKCOPY** . Mais cela oblige, à chaque fois, à relancer la comparaison.

Or, une méthode plus simple existe, c'est le **VERIFY** du DOS. On peut la mettre en service avec **VERIFY ON** , sous DOS :

```
C:\> VERIFY ON

C:\>
```

Vérifier son état avec :

```
C:\> VERIFY
VERIFY est on
```

La mettre hors service :

```
C:\> VERIFY OFF

C:\>
```

Et le vérifier :

```
C:\> VERIFY
VERIFY est off

C:\>
```

Nous vous conseillons d'inclure une ligne **VERIFY ON** dans votre **AUTOEXEC.BAT**. Vous n'aurez plus à y songer et, automatiquement, toutes vos copies seront vérifiées. Le seul inconvénient : les écritures seront ralenties puisque le système les vérifiera, en plus.

17

Une méthode de simple pour assurer la confidentialité de fichiers

Problème

On veut interdire l'accès à un fichier important à tout autre collaborateur de l'entreprise. Comment le faire de façon très simple ?

CONFIDENTIALITE

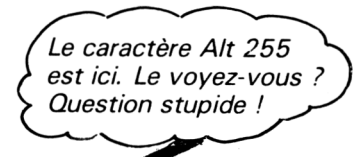
Solution

La méthode la plus simple que nous connaissions consiste à utiliser un caractère « invisible » *à la fin* du nom du fichier. Ce caractère, c'est le tout dernier du code ASCII ; son code est 255. Il se traduit par un espace, mais *ce n'est pas un espace* (l'espace est interdit dans un nom de fichier).

Pour obtenir ce caractère, il faut maintenir la touche *Alt* enfoncée et frapper sur le clavier *numérique* le code 255.

N'utilisez pas les touches numériques du grand clavier alphabétique, car cela ne fonctionnerait pas. Le résultat ne sera pas apparent et quiconque voulant utiliser ce fichier sans connaître l'astuce se verra répondre qu'il n'existe pas !

Voyez ce que cela donne. On a créé un fichier **DUPONT** en frappant, aussitôt derrière le T, un *Alt 255*. Ce pourrait être l'espace situé derrière ce nom, ci-dessus ; mais comment le deviner ? Voici ce fichier :




```
C:\> COPY CON DUPONT
On a utilisé le caractère invisible ALT 255
à la fin du nom "Dupont".
^Z
        1 Fichier(s) copié(s)

C:\>
```

Un utilisateur non averti lira **DUPONT** dans le **DIR** du répertoire, sans se douter qu'il existe un ALT 255. S'il veut l'appeler, l'utiliser ou l'afficher (exemple suivant), il obtiendra :

```
C:\> TYPE DUPONT
Fichier introuvable

C:\>
```

Par contre, si l'on frappe DUPONT ALT 255, tout fonctionne :

```
C:\> TYPE DUPONT
On a utilisé le caractère invisible ALT 255
à la fin du nom "Dupont".

C:\>
```

C'est simple et efficace. Un praticien chevronné, un informaticien, pourront peut-être découvrir le pot-aux-roses, mais tout autre utilisateur ne pourra accéder au fichier et se *« perdra en conjectures et s'égarera en suppositions »* (*Christophe*).

18

Comment interdire l'effacement d'un fichier

INTERDIRE L'EFFACEMENT D'UN FICHER

Problème

L'effacement accidentel d'un fichier précieux est, hélas, chose possible, et ça n'arrive pas qu'aux autres : il suffit d'une manipulation maladroite en fin de journée, quand l'attention est relâchée, ou de l'intervention intempestive d'une autre personne sur votre micro. On peut éviter aisément ce genre d'incident.

Solution

ATTRIB

Avec les DOS des versions 3.*xx*, on peut « marquer » un fichier de façon qu'il ne puisse être que lu : on ne pourra rien écrire, et encore moins l'effacer. La commande est **ATTRIB** , pour *attribut* ; c'est une commande *externe* du DOS, ce qui signifie qu'elle devra être accessible lorsqu'elle sera invoquée (elle devra se trouver dans le répertoire actif, ou dans un répertoire qu'on indiquera *avant* la commande, ou encore dans un répertoire de rechange, spécifié avec un **PATH**). Elle fonctionne ainsi :

+R

· **ATTRIB** suivi de **+R** et du nom du fichier met celui-ci en *lecture* uniquement (R provient de « *Read* », lire).

-R

· **ATTRIB** suivi de **-R** et du nom du fichier le remet en mode « normal », lecture et écriture, ou effacement !

Afin de vous en montrer l'usage, on a créé un fichier appelé **TEST** . On commence par le placer en mode *Lecture* uniquement :

```
C:\> ATTRIB +R TEST

C:\>
```

La commande **ATTRIB** , sans **+R** ni **-R** , mais avec le nom du fichier, permet la vérification de l'*attribution* active ; par curiosité, cela donne :

```
C:\> ATTRIB TEST
  A     R  C:\TEST

C:\>
```

L'attribut est positionné sur R

Tentons d'effacer ce fichier ; le système va le refuser :

```
C:\> DEL TEST
Accès refusé

C:\>
```

Ainsi, ce fichier ne peut être effacé, ni même modifié. Afin de lui ré-attribuer son état primitif, on inverse le signe de R ; on va ensuite vérifier son état, par acquis de conscience, et l'effacer afin de vous prouver que c'est désormais faisable :

```
C:\> ATTRIB -R TEST

C:\>

C:\> ATTRIB TEST
  A        C:\TEST

C:\>

C:\> DEL TEST

C:\>
```

Le fichier a bien été détruit. La commande **ATTRIB** est si facile à employer qu'il serait dommage de s'en priver si l'on tient à la sécurité. Protégez, avant tout, tous vos fichiers *qui n'ont réellement pas à être modifiés*, puis vos fichiers de données sensibles. N'oubliez pas d'en faire des copies, de toutes façons !

19

Comment chaîner des commandes ?

Problème

On veut simultanément commander plusieurs opérations sous DOS. Comment expliquer au système d'exploitation qu'il s'agit de commandes consécutives dont l'une dépend de l'autre, et qu'elles ne sauraient être dissociées ?

Solution

CHAINAGE DE COMMANDES

Le DOS permet un chaînage de commandes grâce à un signe cabalistique, qui est le *trait vertical coupé.* Vous le trouverez dans les listages suivant, car la photocomposition traditionnelle a du mal à l'imprimer !

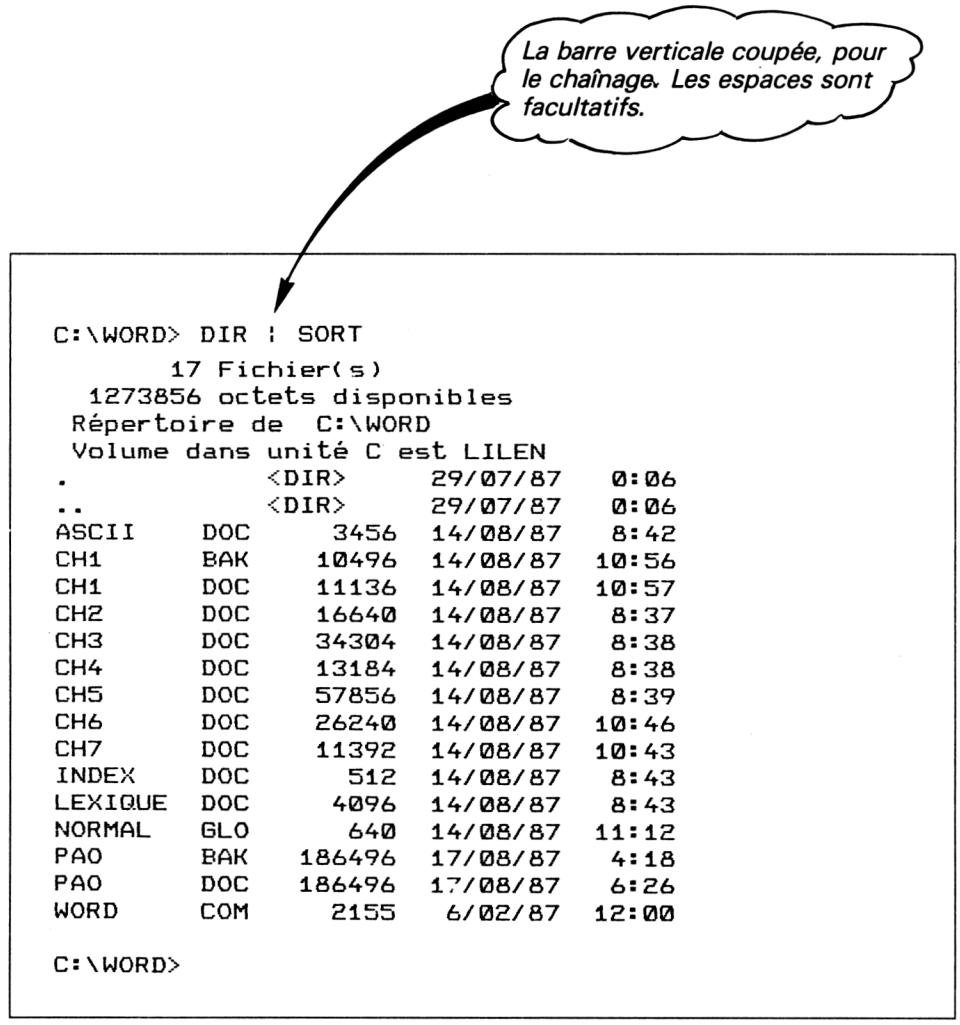
TRAIT VERTICAL COUPE

Le trait vertical coupé est obtenu, sur le clavier :
· En maintenant la touche *Alt* enfoncée.
· Puis, en frappant le code 124 sur le bloc numérique (de droite) du clavier.
Attention : n'essayez pas d'utiliser les touches numériques du clavier principal, alphabétique, car cela ne fonctionnerait pas.

Si vous vous reportez à la table du code ASCII, en fin d'ouvrage, vous constaterez, en effet, que le caractère de code 124 est bien celui du trait vertical coupé.

ALT 124

L'exemple classique du chaînage des commandes est le suivant : on ordonne un DIR pour afficher le répertoire, mais on veut qu'il vienne par ordre alphabétique, ce qui implique l'usage d'une seconde commande, `SORT` . Pour les *chaîner*, on fera intervenir le *Alt 124* ; ainsi, l'auteur a trié son sous-répertoire de traitement de texte WORD :



Faute d'un tel tri, les « chapitres » CH1, CH2, etc., se seraient mélangés, ce qui aurait compliqué la lecture et l'interprétation.

20

Comment ordonner des informations par ordre alphabétique, ou numérique, ou mixte ?

Problème

On a créé un fichier contenant des informations qu'on souhaite classer par ordre numérique ou alphabétique. Par exemple, un fichier de noms et d'adresses, ou de références, etc. Le DOS en offre-t-il le moyen ?

Solution

CLASSEMENT

SORT

Caractéristique généralement fort méconnue, le DOS offre une méthode de classement qui, de surcroit, est aussi efficace que rapide. L'auteur l'utilise pour des applications professionnelles de gestion de fichiers (et leur édition), de préférence à toute autre. Et puis, elle est tellement simple à appliquer ! Elle implique l'usage de la commande `SORT` et, pour vous en faire la démonstration, on va lister par ordre alphabétique les commandes classiques du DOS.

Tout d'abord, sans `SORT` , le `DIR` fournit une liste qui, pour l'auteur et en abrégé, est la suivante :

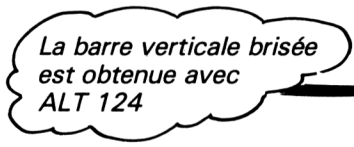
```
C:\DOS> DIR/W

 Volume dans unité C est LILEN
 Répertoire de  C:\DOS

.                  ..                 ØBARRE   BAS     NUMERIQU COM     ANSI     SYS
COUNTRY  SYS       DISPLAY  SYS       DRIVER   SYS     FASTOPEN EXE     FDISK    COM
FORMAT   COM       KEYB     COM       KEYBOARD SYS     MODE     COM     NLSFUNC  EXE
PRINTER  SYS       REPLACE  EXE       SELECT   COM     SYS      COM     VDISK    SYS
XCOPY    EXE       EGA      CPI       LCD      CPI     4201     CPI     5202     CPI
NUM      COM       APPEND   EXE       ASSIGN   COM     ATTRIB   EXE     BACKUP   COM
BASIC    COM       BASICA   COM       CHKDSK   COM     COMP     COM     DEBUG    COM
DISKCOMP COM       DISKCOPY COM       EDLIN    COM     FIND     EXE     GRAFTABL COM
GRAPHICS COM       JOIN     EXE       LABEL    COM     MORE     COM     PRINT    COM
RECOVER  COM       RESTORE  COM       SHARE    EXE     SORT     EXE     SUBST    EXE
TREE     COM       BASIC    PIF       BASICA   PIF     MORTGAGE BAS
       54 Fichier(s)
  1302528 octets disponibles

C:\DOS>
```

Tout est dans le désordre alphanumérique. Maintenant, on va demander un **DIR** avec classement (**SORT**) par ordre alphanumérique croissant. *On va donc combiner, chaîner, deux commandes.*

Pour chaîner des commandes, on les sépare par une barre verticale brisée. Elle est obtenue en maintenant la touche *Alt* enfoncée et en frappant 124 *sur le clavier numérique* (de droite).

Attention : n'utilisez pas les autres touches numériques car cela ne fonctionnerait pas ! Le système va alors prendre un instant pour organiser le fichier selon l'ordre fourni par la table des codes ASCII, c'est-à-dire les chiffres d'abord. Voici ce que cela procure sur le même répertoire ; remarquez que tout ce qui commence par un chiffre vient en premier (voir page suivante).

C'est donc beaucoup mieux ainsi. Mais on peut aller au-delà, encore.




```
C:\DOS> DIR ¦ SORT


      54 Fichier(s)
  1298432 octets disponibles
 Répertoire de  C:\DOS
 Volume dans unité C est LILEN
.             <DIR>      17/04/87    9:22
..            <DIR>      17/04/87    9:22
ØBARRE   BAS        3Ø    1/Ø1/8Ø    1:43
4201     CPI    17Ø89   18/Ø3/87   12:ØØ
5202     CPI      459   18/Ø3/87   12:ØØ
ANSI     SYS     1678   18/Ø3/87   12:ØØ
APPEND   EXE     5873   18/Ø3/87   12:ØØ
ASSIGN   COM     1566   18/Ø3/87   12:ØØ
ATTRIB   EXE     9551   18/Ø3/87   12:ØØ
BACKUP   COM    31961   18/Ø3/87   12:ØØ
BASIC    COM     1Ø75   18/Ø3/87   12:ØØ
BASIC    PIF      369   18/Ø3/87   12:ØØ
BASICA   COM    364Ø3   18/Ø3/87   12:ØØ
BASICA   PIF      369   18/Ø3/87   12:ØØ
CHKDSK   COM    1Ø282   18/Ø3/87   12:ØØ
COMP     COM     4294   18/Ø3/87   12:ØØ
COUNTRY  SYS    11285   18/Ø3/87   12:ØØ
DEBUG    COM    15948   18/Ø3/87   12:ØØ
DISKCOMP COM     5975   18/Ø3/87   12:ØØ
DISKCOPY COM     6439   18/Ø3/87   12:ØØ
DISPLAY  SYS    11316   18/Ø3/87   12:ØØ
DRIVER   SYS     1192   18/Ø3/87   12:ØØ
EDLIN    COM     767Ø   18/Ø3/87   12:ØØ
EGA      CPI    49Ø65   18/Ø3/87   12:ØØ
FASTOPEN EXE     3967   18/Ø3/87   12:ØØ
FDISK    COM    48686   18/Ø3/87   12:ØØ
FIND     EXE     6455   18/Ø3/87   12:ØØ
FORMAT   COM    11968   18/Ø3/87   12:ØØ
GRAFTABL COM     6122   18/Ø3/87   12:ØØ
GRAPHICS COM     33Ø3   18/Ø3/87   12:ØØ
JOIN     EXE     8995   18/Ø3/87   12:ØØ
KEYB     COM     912Ø   18/Ø3/87   12:ØØ
KEYBOARD SYS    19766   18/Ø3/87   12:ØØ
LABEL    COM     2472   18/Ø3/87   12:ØØ
LCD      CPI    1Ø752   18/Ø3/87   12:ØØ
MODE     COM    15791   18/Ø3/87   12:ØØ
MORE     COM      315   18/Ø3/87   12:ØØ
MORTGAGE BAS     6251   18/Ø3/87   12:ØØ
NLSFUNC  EXE     3Ø73   18/Ø3/87   12:ØØ
NUM      COM       17    1/Ø1/8Ø    8:53
NUMERIQU COM       15   16/Ø6/87    3:14
PRINT    COM     925Ø   18/Ø3/87   12:ØØ
PRINTER  SYS    13622   18/Ø3/87   12:ØØ
RECOVER  COM     4427   18/Ø3/87   12:ØØ
REPLACE  EXE    11983   18/Ø3/87   12:ØØ
RESTORE  COM    348Ø3   18/Ø3/87   12:ØØ
SELECT   COM     43Ø7   18/Ø3/87   12:ØØ
SHARE    EXE     8656   18/Ø3/87   12:ØØ
SORT     EXE     1993   18/Ø3/87   12:ØØ
SUBST    EXE     9931   18/Ø3/87   12:ØØ
SYS      COM     483Ø   18/Ø3/87   12:ØØ
TREE     COM     3586   18/Ø3/87   12:ØØ
VDISK    SYS     3579   18/Ø3/87   12:ØØ
XCOPY    EXE    11486   18/Ø3/87   12:ØØ
```

Tris directs et inverses, ou sur colonne

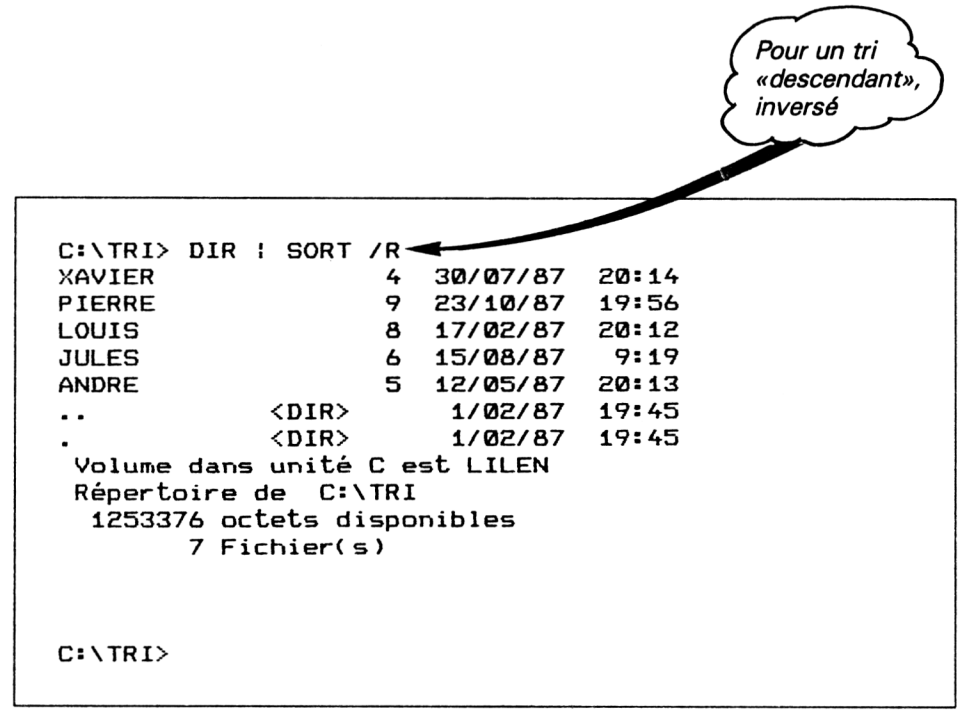
Pour la suite de cette démonstration, on a créé un court fichier dans le sous-répertoire TRI. Il se présente ainsi :

```
C:\TRI> DIR

 Volume dans unité C est LILEN
 Répertoire de  C:\TRI

.            <DIR>       1/02/87  19:45
..           <DIR>       1/02/87  19:45
PIERRE                9  23/10/87  19:56
LOUIS                 8  17/02/87  20:12
ANDRE                 5  12/05/87  20:13
XAVIER                4  30/07/87  20:14
JULES                 6  15/08/87   9:19
        7 Fichier(s)
  1257472 octets disponibles

C:\TRI>
```

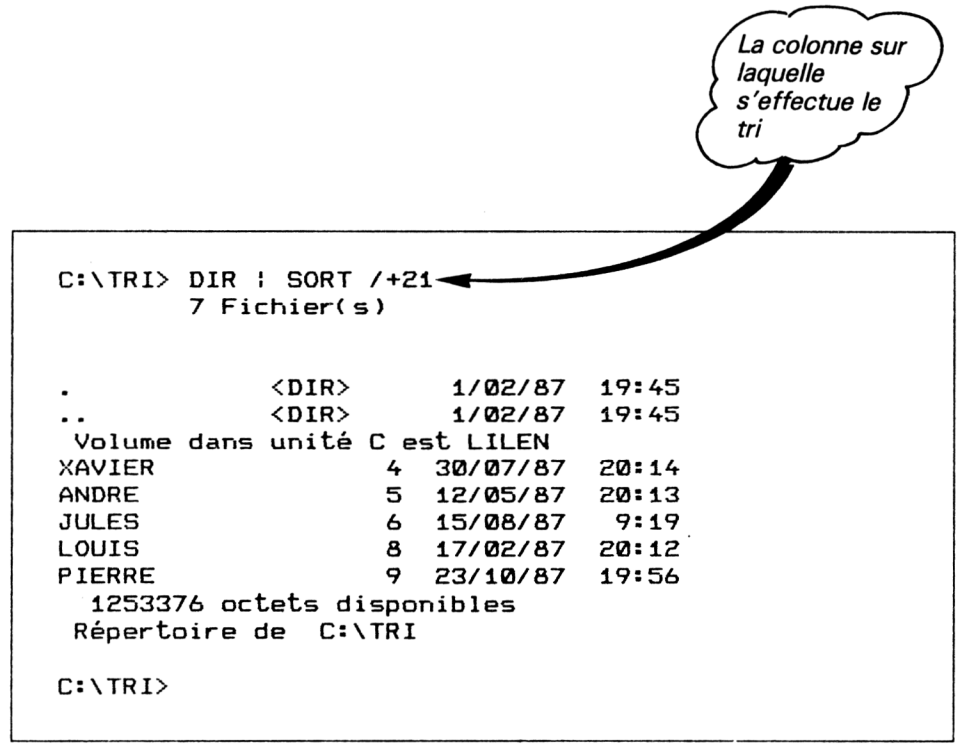
Avec un tri comme précédemment, donc *ascendant*, on obtient :

```
C:\TRI> DIR | SORT


        7 Fichier(s)
  1253376 octets disponibles
 Répertoire de  C:\TRI
 Volume dans unité C est LILEN
.            <DIR>       1/02/87  19:45
..           <DIR>       1/02/87  19:45
ANDRE                 5  12/05/87  20:13
JULES                 6  15/08/87   9:19
LOUIS                 8  17/02/87  20:12
PIERRE                9  23/10/87  19:56
XAVIER                4  30/07/87  20:14

C:\TRI>
```

On peut trier dans l'ordre *inverse*, en ajoutant la clé **/R** (avec R pour « *Reverse* ») :



Ici, on commence par *Xavier* pour finir par *André*. Sans autre mention, le tri s'est, jusqu'ici, effectué par lignes, en commençant par le début de chacune d'elles ; maintenant, *on va spécifier une colonne*. Si, par exemple, on veut trier sur *la longueur des fichiers* (*Xavier* contient 4 caractères, *Pierre* 9, etc.), et parce que ces chiffres sont situés dans la 21e colonne, on commandera un « **SORT /+21** ». N'oubliez pas le symbole *plus* (**+**) ; on obtient :



Sur le même principe, on peut trier sur les mois, lesquels commencent en colonne 26 :

```
C:\TRI> DIR | SORT /+26

 Répertoire de  C:\TRI

        7 Fichier(s)
.             <DIR>       1/02/87  19:45
..            <DIR>       1/02/87  19:45
LOUIS                 8  17/02/87  20:12
ANDRE                 5  12/05/87  20:13
XAVIER                4  30/07/87  20:14
JULES                 6  15/08/87   9:19
PIERRE                9  23/10/87  19:56
  1253376 octets disponibles
 Volume dans unité C est LILEN

C:\TRI>
```

Ou encore, sur les heures (colonne 31) :

```
C:\TRI> DIR | SORT /+31
 Volume dans unité C est LILEN

 Répertoire de  C:\TRI

        7 Fichier(s)
  1253376 octets disponibles
JULES                 6  15/08/87   9:19
.             <DIR>       1/02/87  19:45
..            <DIR>       1/02/87  19:45
PIERRE                9  23/10/87  19:56
LOUIS                 8  17/02/87  20:12
ANDRE                 5  12/05/87  20:13
XAVIER                4  30/07/87  20:14

C:\TRI>
```

On pourrait encore inverser tout cela, avec **/R** . La seule limite au tri réside dans la longueur du fichier, avec un maximum de 64 Ko. Même dans ce cas, l'agorithme de tri retenu par *Microsoft* se révèle l'un des plus rapides qui soient. Ajoutons que le DOS attribue aux caractères accentués les codes du même caractère, sans accent, mais en majuscule.

21

Comment trier les informations à afficher ou imprimer ?

Problème

On veut afficher la seule liste des fichiers contenant le mot CLIENTS. Ou la liste des seuls noms de sous-répertoires.

Solution

FILTRE

RECHERCHER UNE CHAINE DE CARACTERES

FIND

BARRE VERTICALE COUPEE

ALT 124

Le DOS possède une commande, dite de « filtre », qui sert à rechercher une chaîne de caractères. C'est `FIND` , pour « trouver ». En voici un exemple : on veut ne retenir que les noms des sous-répertoires figurant dans le répertoire principal, supposé contenir de tout. Seuls, les noms des sous-répertoires seront affichés. Pour cela, on remarque que la ligne des noms de sous-répertoires comporte la mention `<DIR>` ; c'est donc elle qu'on va rechercher et ce sont uniquement les lignes incluant cette mention qui seront retenues (« filtrées »).

La commande s'effectue en chaînant `DIR` et `FIND` à l'aide du caractère « barre verticale coupée ».

Le caractère « barre verticale coupée » est obtenu en maintenant *Alt* enfoncée, puis en frappant 124 sur le clavier numérique exclusivement.

N'utilisez pas les autres touches numériques, ça ne marcherait pas.

D'autre part, *la chaîne à rechercher doit figurer entre guillemets*, impérativement.
Pour une première démonstration, l'auteur est passé au sous-répertoire ALPHA qui contient : un sous-sous-répertoire BETA, plus la marque du répertoire père (noté à l'aide des deux points en ligne), plus sa propre référence en tant que sous-répertoire (un unique point). Voici l'ordre complet et son résultat :

ALT 124

```
C:\ALPHA> DIR ¦ FIND "<DIR>"
.             <DIR>      31/08/87    2:14
..            <DIR>      31/08/87    2:14
BETA          <DIR>      31/08/87    2:14

C:\ALPHA>
```

En fait, on peut se contenter de rechercher le symbole « **<** », qui suffit ici :

```
C:\ALPHA> DIR ¦ FIND "<"
.             <DIR>      31/08/87    2:14
..            <DIR>      31/08/87    2:14
BETA          <DIR>      31/08/87    2:14

C:\ALPHA>
```

Voici un second exemple pratique. Dans son sous-répertoire WINDOWS, l'auteur dispose d'un certain nombre de polices de caractères, des *fontes*. Leur nom est doté de l'extension **.FON** . Pour en dresser la liste, il suffit de commander :

```
C:\WINDOWS> DIR ¦ FIND "FON"
COURA     FON      8720   3/12/86   14:03
HELVA     FON     27056   3/12/86   14:03
TMSRA     FON     26256   3/12/86   14:03
COURD     FON     15136   3/12/86   14:03
HELVD     FON     45600   3/12/86   14:03
TMSRD     FON     45200   3/12/86   14:03
COURC     FON      8784   3/12/86   14:03
HELVC     FON     26624   3/12/86   14:03
TMSRC     FON     25824   3/12/86   14:03
ROMAN     FON     11120   3/12/86   14:03
SCRIPT    FON     10304   3/12/86   14:03
MODERN    FON      7584   3/12/86   14:03
COURB     FON     12304   3/12/86   14:03

C:\WINDOWS>
```

D'un seul coup d'œil, on sait quelles fontes sont disponibles.

22

Structure de DOS sur un disque : ce qu'est la FAT

Problème

On entend souvent parler de la FAT, sur un disque, en notant combien elle est importante. Mais qu'est-ce, au juste ?

Solution, ou plutôt réponse

Sur un disque (et sans que le système soit chargé), le DOS se réserve un espace minimal comprenant :

* *Une amorce* : s'il s'agit d'un disque système, c'est elle qui appellera le DOS. Sinon, elle déclenchera un message d'erreur si le DOS n'a pas été préalablement chargé.
* *Une table d'attribution des fichiers*, ou FAT, de *« File Allocation Table »*. Elle tient à jour le plan d'occupation du disque.

TABLE D'ATTRIBUTION DES FICHIERS

FAT

L'unité d'occupation n'est pas le secteur, mais le « bloc » (*« cluster »*, en anglais), comprenant plusieurs secteurs. Leur nombre réel dépend du DOS.

CLUSTER

· Un *cluster* représente un espace *logique* et non *physique*. Par exemple :

· Sur une disquette double face, un *cluster* représente 2 secteurs soit 1024 octets.
· Sur le disque dur 10 Mo de l'XT, un cluster vaut 8 secteurs, soit 4096 octets.
· Sur le disque 20 Mo de l'AT2, le cluster représente 4 secteurs, soit 2048 octets.

Ce phénomène explique l'incompatibilité apparente entre les divers DOS, appliqués au disque fixe.

Cette FAT étant très importante, elle est généralement doublée (sans que, d'ailleurs, sa doublure intervienne !).
Sa longueur dépend du nombre de *clusters*, donc du DOS.

REPERTOIRE RACINE

REPERTOIRE PRINCIPAL

* *Le 3e élément d'entrée, sur le disque, est le répertoire racine*, ou *répertoire principal.* Il enregistre les noms des fichiers, avec leurs extensions (sur 8 + 3 caractères), la longueur du fichier, la date et l'heure de la création ou de la modification, mais aussi le *numéro d'ordre du premier cluster de chaque fichier* et *l'attribut du fichier* : fichier système, fichier caché, fichier à lecture uniquement. Chaque entrée dans un répertoire occupe 32 octets ; de ce fait, 16 entrées trouvent place dans un secteur.

La suite de ces informations peut être illustrée comme le montre la figure 1.

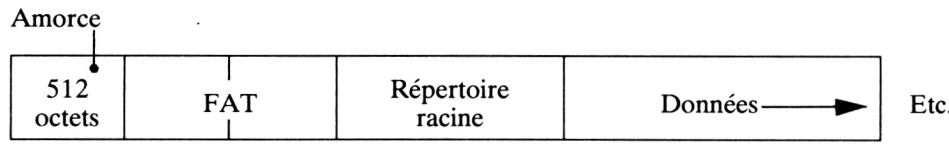


Fig. 1. — Les premières informations sur un disque.

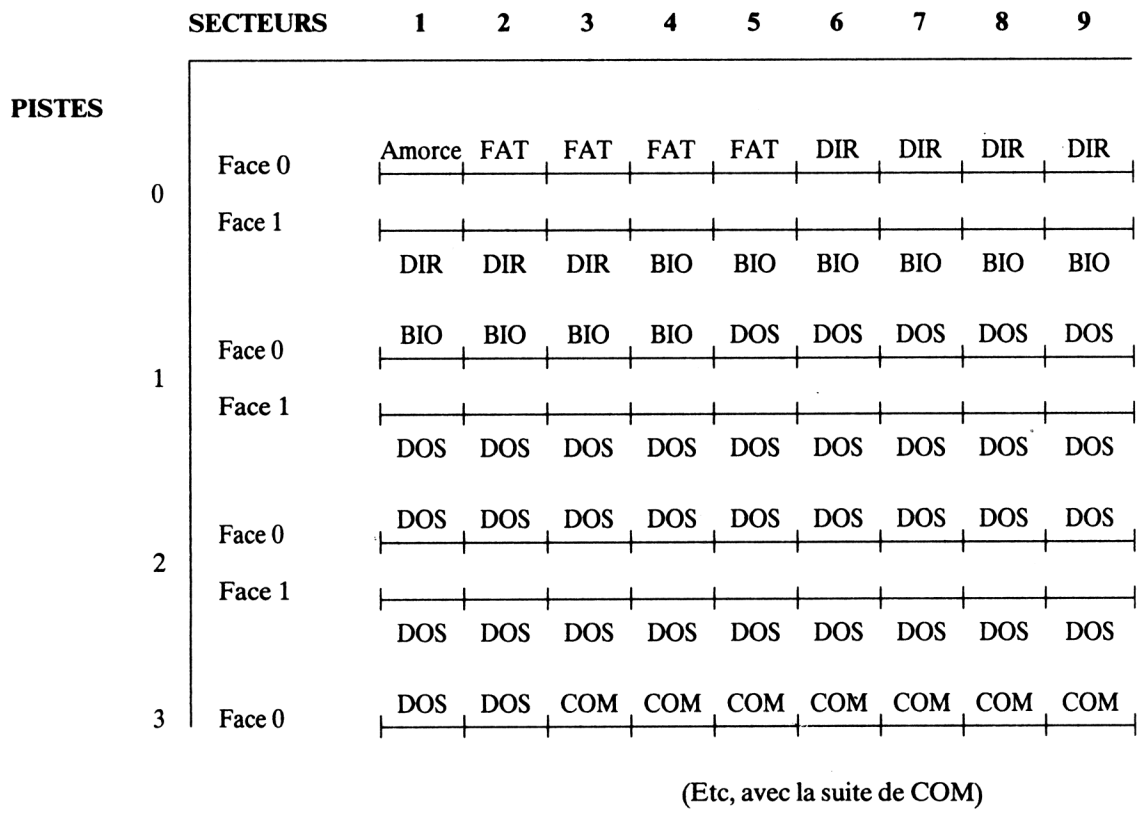
A titre d'illustration, la figure 2 montre le démarrage d'une disquette 360 Ko, double face, à 9 secteurs par piste. On y retrouve l'amorce et la FAT, suivies par le répertoire principal (DIR) puis, et dans le cas d'une disquette système, ici, par les zones réservées au DOS.

Revenons à la FAT. C'est une table qui se borne à classer les clusters par numéros d'ordre. Si un numéro est codé sur 2 octets, on pourra couvrir jusqu'à 64 K clusters en utilisant une table de 64 K positions (à 2 octets), occupant donc 64 K × 2 = 128 Ko, soit 256 secteurs.

Supposons un disque 2Ø Mo avec des clusters de 4 secteurs, donc 2 Ko ; il suffira, en fait, de 2Ø Mo/2 Ko = 1Ø K clusters ; la FAT pourra donc être limitée.

· Les positions Ø et 1 de la FAT ne sont pas utilisées pour numéroter les clusters. Elles servent à enregistrer un identificateur servant au DOS à illustrer le format du disque.
· Un Ø signifie que le cluster est libre.
· Un code FFFF (pour des codes sur 16 bits) marque la fin d'un fichier.
· Les codes FFFØ à FFFF (en 16 bits) sont réservés.

Rappelez-vous que le **répertoire racine** fournit le numéro du premier cluster occupé par un fichier. Supposons qu'il s'agisse du numéro 4. On va entrer dans la FAT à la position 4 ; celle-ci fournit le numéro du cluster suivant, et ainsi de suite, en chaîne, jusqu'au FFFF (fig. 3).



BIO = IBMBIO. COM pour IBM, ou 10 SYS pour Microsoft
DOS = IBMDOS. COM pour IBM, ou MSDOS SYS pour Microsoft

Fig. 2. — Organisation d'une disquette 360 K

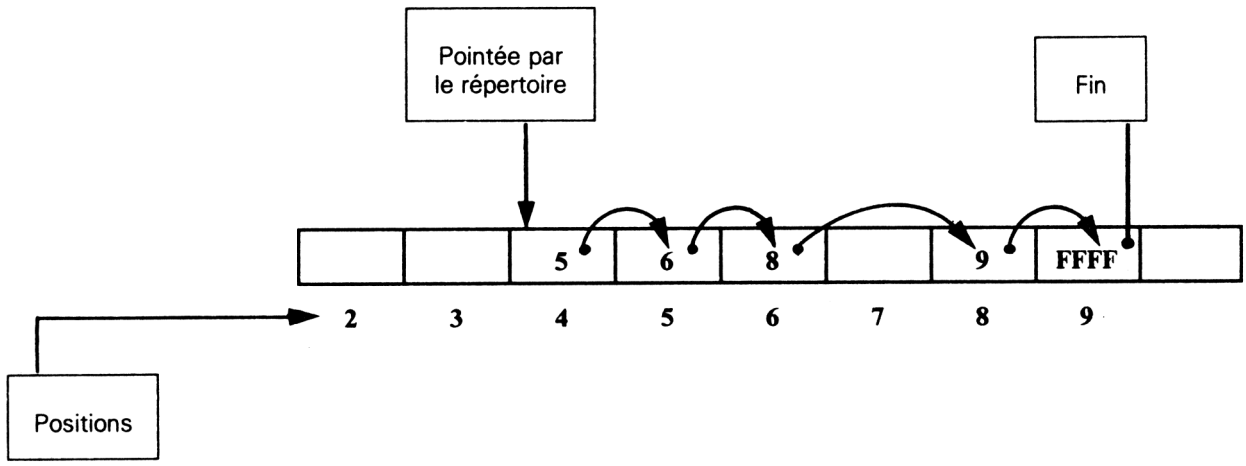


Fig. 3. — Principe de fonctionnement de la FAT. Le répertoire racine pointe la première position. Celle-ci renvoie au « cluster » (bloc) suivant, et ainsi de suite jusqu'à la marque de fin de fichier.

Notez que chaque position occupe 16 bits ici, mais qu'avec les disques faible capacité, elle peut être sur 12 bits, ce qui permet le codage de 4 K clusters. Si un cluster représente 8 secteurs, soit 4 K, cela couvrira encore 16 Mo.

Enfin, et à propos des répertoires, ajoutons ces quelques précisions. Chaque entrée est sur 32 octets et contient (fig. 4) :

- *Le nom du fichier*, sur 8 octets.
- *Son extension*, sur 3 octets.
- *La taille* du fichier, codée sur 4 octets (un nombre entier).
- *La date*, sur 2 octets.
- *L'heure*, sur 2 octets.
- *Le numéro de cluster de départ*, sur 2 octets.
- *Un attribut*, sur 1 octet.
- *Dix octets libres*, et réservés.

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td colspan="8">Nom du Fichier</td></tr>
<tr><td>8</td><td>9</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td></tr>
<tr><td colspan="2">Extension</td><td>Attribut</td><td colspan="5">Réservé</td></tr>
<tr><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td></tr>
<tr><td colspan="6">Réservé</td><td colspan="2">Heure de création ou mise à jour</td></tr>
<tr><td>18</td><td>19</td><td>1A</td><td>1B</td><td>1C</td><td>1D</td><td>1E</td><td>1F</td></tr>
<tr><td colspan="2">Date de création ou de mise à jour</td><td colspan="2">Numéro du cluster de départ</td><td colspan="4">Taille du fichier
Mot bas | Mot haut
octet bas | octet haut | octet bas | octet haut</td></tr>
</table>

Fig. 4. — Entrée type dans un répertoire, sur 32 octets (0 à 1 F, en hexadécimal).

Si le nom du fichier commence par un :
· 0, c'est que l'entrée n'a encore jamais été utilisée.
· Le caractère ASCII E5, c'est que le fichier a été effacé.

L'effacement d'un fichier se traduit donc par la seule introduction de ce caractère dans le répertoire ; en réalité, son contenu existe toujours mais il n'est plus accessible. Cela explique que certains programmes, tels les « **Utilitaires de Norton** », permettent la récupération de fichiers malencontreusement effacés (à la condition qu'on n'ait pas ré-écrit de nouvelles informations à leur place).

Les date et heure sont codées à partir des formules de calcul suivantes :

DATE = JOUR + 32 * MOIS + 512 * (ANNEE − 1980)
(de 1980 à 2099)

HEURE = SECONDES/2 + 32 * MINUTES + 1024 * HEURES

Enfin, *l'attribut* marque les entrées pour un label de disque, un sous-répertoire, un fichier à lecture exclusivement, une modification de fichier (qui sera interrogé par **BACKUP** , du DOS), un fichier système ou un fichier caché. Les fichiers systèmes et cachés (ces deux catégories sont pratiquement semblables) restent invisibles pour les ordres du DOS tels que **DEL, COPY, DIR,** etc. Ces codes s'ajoutent lorsque les attributs sont cumulés.

ATTRIBUTS DES FICHIERS

Valeur (binaire)	Signification
0000 0001	Fichier marqué à lecture uniquement. Peut se conjuguer avec les autres codes
0000 0010	Fichier caché
0000 0100	Fichier système
0000 1000	Contient un label de volume, sur 11 caractères, sans aucune autre information. Ne peut exister que dans le répertoire racine.
0001 0000	Définit un sous-répertoire
0010 0000	Archivage. Positionné à 1 lorsque le fichier a été écrit et refermé. Utilisé par BACKUP et RESTORE

On peut aisément lire ces entrées avec **DEBUG** . Sur une disquette 360 Ko, on appellera **DEBUG** et on commandera : **L 100 0 5 7**
C'est l'ordre de charger à partir de l'offset 1ØØ, de l'unité à disquettes Ø (c'est la **A:**), à partir du secteur 5, les 7 suivants (donc, jusqu'à **0BH**). Telle est, en effet, la position du répertoire sur une disquette 360 Ko. Puis, on frappe : **d**

On affiche alors le contenu lu, en un premier écran. Des frappes répétées de **d** afficheront les écrants suivants. Un processus complet, avec un seul écran et portant sur une disquette DOS, montrerait :

```
A:\>C:DEBUG
-L 100 0 5 7
-D
1157:0100  49 42 4D 42 49 4F 20 20-43 4F 4D 27 00 00 00 00   IBMBIO  COM'....
1157:0110  00 00 00 00 00 00 AC 3B-8B 0A 02 00 78 25 00 00   ......¬;....x%..
1157:0120  49 42 4D 44 4F 53 20 20-43 4F 4D 27 00 00 00 00   IBMDOS  COM'....
1157:0130  00 00 00 00 00 00 83 62-96 0A 0C 00 70 6C 00 00   .......b....pl..
1157:0140  43 4F 4D 4D 41 4E 44 20-43 4F 4D 20 00 00 00 00   COMMAND COM ....
1157:0150  00 00 00 00 00 00 00 60-96 0A 28 00 7A 5C 00 00   .......`..(.z\..
1157:0160  41 53 53 49 47 4E 20 20-43 4F 4D 20 00 00 00 00   ASSIGN  COM ....
1157:0170  00 00 00 00 00 00 00 60-96 0A 40 00 EA 05 00 00   .......`..@.....
-Q

A:\>
```

On trouve, bien sûr, les fichiers **IBMBIO** et **IBMDOS** en tout premier lieu. Leur attribut est **27H** (voyez la 12e colonne des octets), ce qui signifie, si on transpose cela en binaire, soit **ØØ1Ø Ø111** , qu'il s'agit à la fois de fichiers à lecture uniquement, cachés, systèmes, et ne devant pas être sauvegardés par un **BACKUP** . La séquence se termine par un **Q** , pour quitter **DEBUG** .

23

Comment copier d'une disquette à l'autre sur un micro avec une seule unité à disquettes ?

Problème

On dispose d'une seule unité à disquettes, en plus du disque dur. Faut-il obligatoirement passer par une copie sur le disque dur pour dupliquer des disquettes ou recopier certains fichiers ?

Solution

DISKCOPY

COPY

XCOPY

Il s'agit là d'un problème qui se pose typiquement aux néophytes. Voici quelques-unes des méthodes utilisables :

- On fait un `DISKCOPY`, de disquette à disquette.
- On utilise `COPY`, de disquette à disquette, ou éventuellement, en passant par le disque dur.
- Avec les DOS à partir des *3.xx*, on utilise `XCOPY`

Voyons cela plus en détail, en notant, tout d'abord, un phénomène important :

Lorsqu'il n'existe qu'une seule unité à disquettes, le système la reconnaîtra sous le nom d'unité `A:` ou d'unité `B:`.

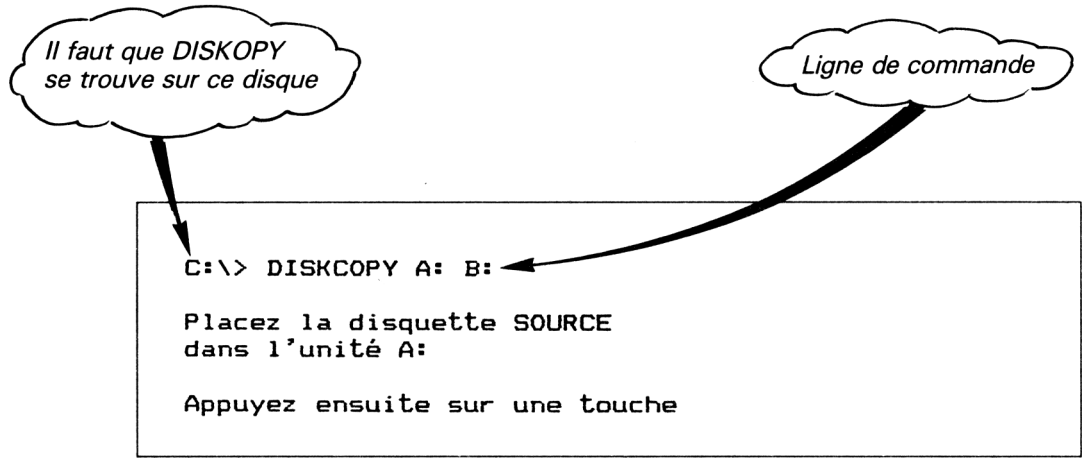
C'est ce qui va nous permettre de travailler *de disquette à disquette.*

1. AVEC DISKCOPY

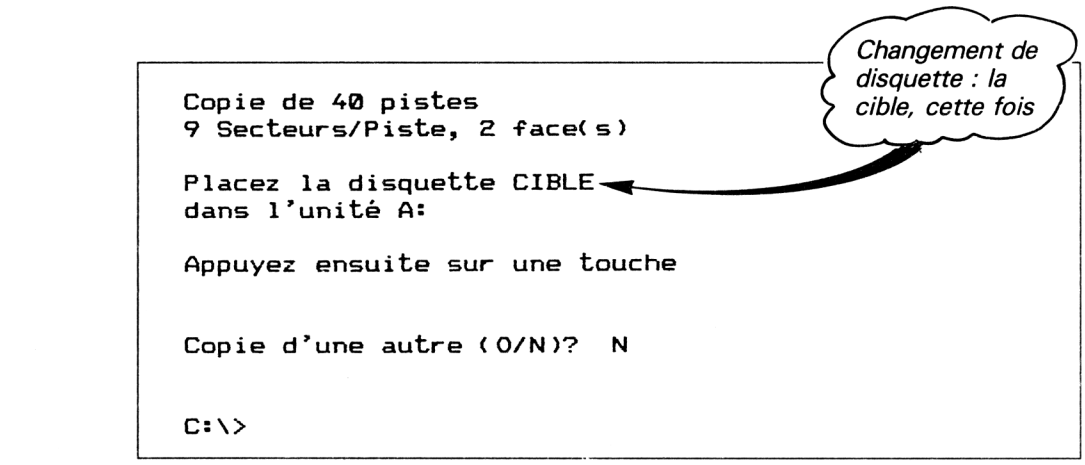
La commande **DISKCOPY** *est une commande* **externe** du DOS ; cela signifie simplement qu'elle est listée dans un **DIR**, et qu'elle doit être présente dans le répertoire actif pour pouvoir être lancée ; ou alors, il faudra spécifier son répertoire. Elle procède à la duplication d'une disquette *à l'identique*, piste pour piste, et secteur pour secteur.

En dupliquant :
· Elle efface ce qui pouvait exister sur la disquette *cible*, celle qui reçoit la copie.
· Si celle-ci n'a pas été formatée, elle la formate au préalable.

La disquette *source*, celle qu'on veut copier, doit être insérée dans l'unité à disquettes, d'abord appellée unité **A:** puis, on va commander un **DISKCOPY** de **A:** sur **B:** en sachant qu'il s'agit *de la même* unité à disquettes. Le système ne s'y trompera toutefois pas car voici ce qui va se passer. D'abord, on frappe la ligne de commande ; en réponse, **DISKCOPY** affiche :



Cela fait, on frappe sur une touche quelconque ; le système va annoncer ce qu'il est en train de faire. Si votre mémoire *centrale* est suffisante, il va lire la disquette source en totalité, ce qui procurera le dialogue complet suivant à l'écran :



Si la mémoire *centrale* est insuffisante, l'écran vous proposera à plusieurs reprises d'interchanger les disquettes.

C'est rapide et efficace ; éventuellement utilisez **DISKCOMP**, de la même façon et après la copie, afin de comparer les disquettes et de vérifier que la copie est aussi bonne que l'original.

2. AVEC COPY

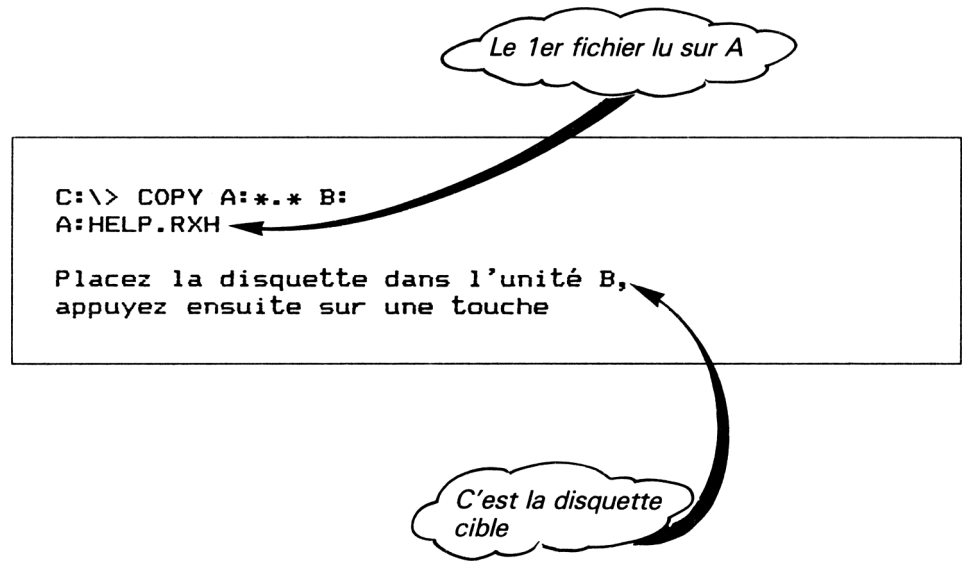
La commande **DISKCOPY** présente un défaut majeur : elle ne tient pas compte d'éventuels secteurs défaillants sur la disquette cible. C'est pourquoi nous vous conseillons l'usage de **DISKCOMP** après un **DISKCOPY** . Même si la disquette cible est parfaite, cette commande offre un second inconvénient : elle recopie les fichiers *à l'identique* ; si ceux-ci sont « morcelés » sur la disquette source (fragmentés, ils ne se suivent pas sur des secteurs consécutifs), on les retrouvera morcelés sur la cible, ce qui réduit la vitesse de lecture. Toutes raisons pour lesquelles nous préférons l'ordre **COPY**.

· COPY procède à la recopie, fichier par fichier.
· Il les ré-organise en resoudant des fragments disjoints.
· Il reconnaît de mauvais secteurs sur la disquette cible et ne les utilise pas.

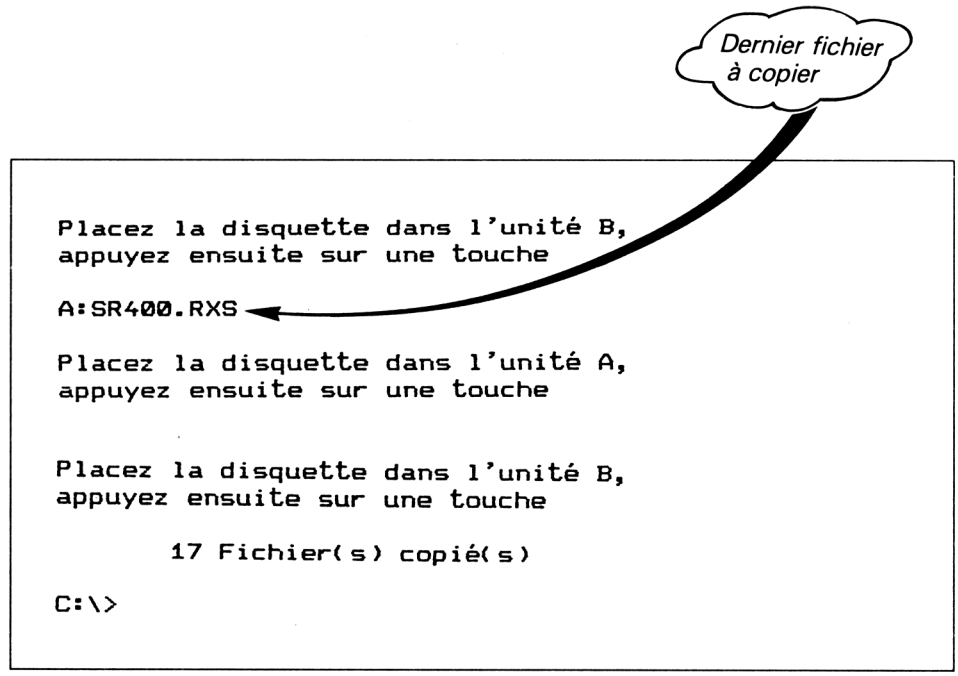
On peut appliquer **COPY** (programme *interne* du DOS, on n'a plus à se préoccuper de sa présence ou de son répertoire) tout comme **DISKCOPY** avec, pour copier *tous* les fichiers :

```
C:\> COPY A:*.* B:
```

Supposons qu'on ait une disquette comportant 17 fichiers à copier. **COPY** va les copier un par un, demandant à chaque fois qu'on change de disquette et affichant, au passage, le nom du fichier en cours de copie :



Si le fichier est long, l'écran vous ordonnera d'intervertir les disquettes ; puis, le système passera au second fichier, de la même façon, puis au troisième, etc., jusqu'à la fin :



La copie est, certes, exécutée, mais il aura fallu intervertir au moins 2 fois 17 = 34 fois les disquettes, ce qui lasse les utilisateurs les plus patients.

· Si le nombre de fichiers à dupliquer est élevé ;
· Si leur nombre est grand ;
· Si l'on veut créer plusieurs disquettes cibles (plusieurs copies),
mieux vaux alors passer par l'intermédiaire du disque dur.

L'auteur utilise un sous-répertoire spécial, appelé TRANSFER, dans lequel il commence par recopier la disquette source avec un :

```
C:\TRANSFER> COPY A:*.*
```

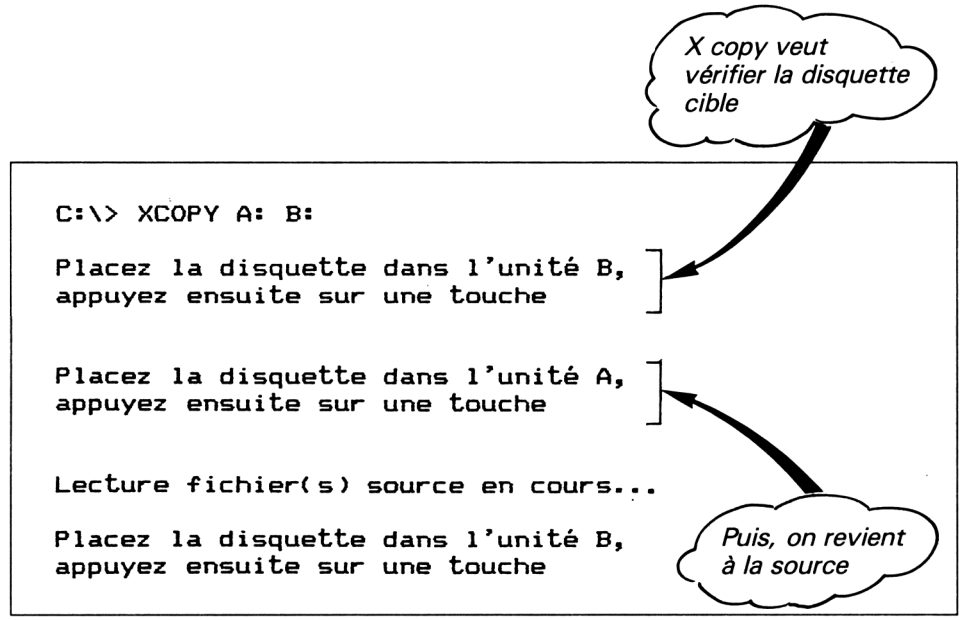
Puis, il recopie le contenu de ce sous-répertoire sur une disquette cible, en `A:`

```
C:\TRANSFER> COPY *.* A:
```

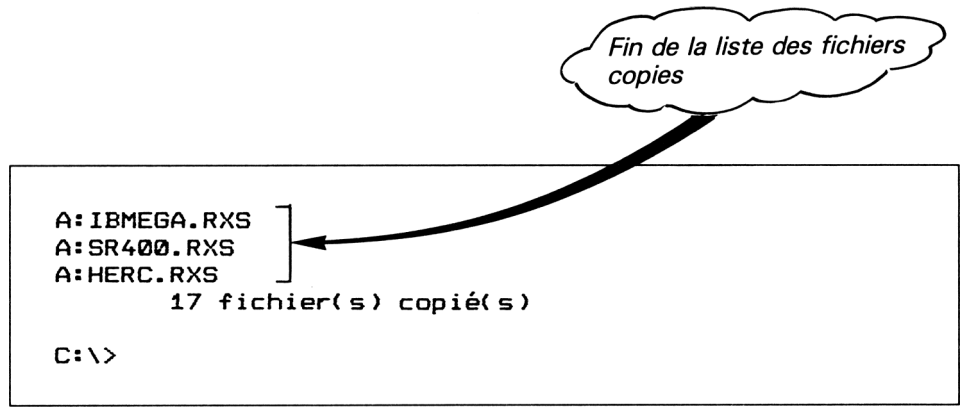
Cela, autant de fois qu'on désire de disquettes cibles.

3. AVEC XCOPY

La commande **XCOPY** , apparue avec les versions 3.xx du DOS, reprend les qualités de **DISKCOPY** et de **COPY** en abandonnant leurs défauts : la copie se fait comme avec **COPY** , mais en salve comme avec **DISKCOPY** . En effet, **XCOPY** lit autant de fichiers de la disquette source que la mémoire *centrale* peut en contenir, puis recopie le tout sur la cible. La disquette *source* étant en A, on fait :



Notez que A est mis pour la source, et B pour la cible ; on ne change que *deux fois* les disquettes, si la mémoire centrale est suffisante ; **XCOPY** affiche les noms des fichiers en cours de copie, puis termine avec :



C'est la formule que nous préférons. *Attention*, toutefois : **XCOPY** est, également, un programme *externe* du DOS (tout comme **DISKCOPY**) ; aussi, le système devra-t-il pouvoir y accéder.

Une autre idée : pour sauvegarder certains fichiers préférez **XCOPY** à **BACKUP** , lorsque cela est possible.

24

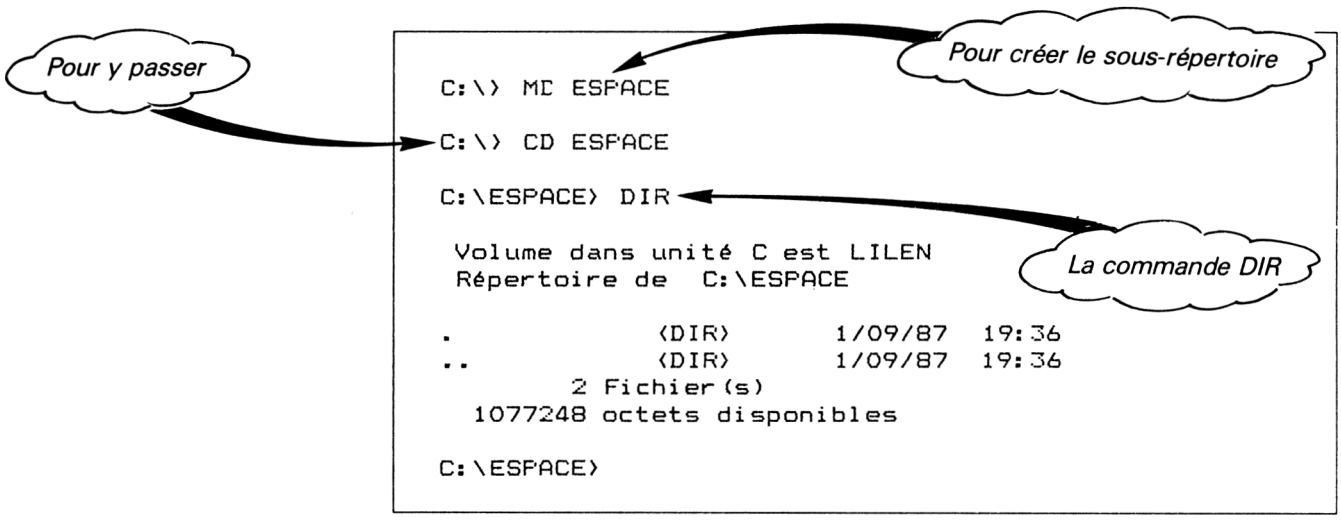
Comment afficher la capacité encore disponible sur le disque dur, et rien d'autre

Problème

Lorsqu'on veut connaître l'espace encore disponible sur le disque dur, le plus simple consiste à faire un **DIR** *. Mais dans ce cas, tout le répertoire défile ! Peut-on court-circuiter cet affichage, inutile dans ce cas ?*

Solution

La solution ressemble fort à celle de l'œuf de Christophe Colomb : il suffit de faire un **DIR** à partir d'un sous-répertoire vide ! Si l'on a créé, sur le disque fixe, un sous-répertoire ESPACE *qui ne servira qu'à cela*, on pourra obtenir directement la capacité du disque qui reste à disposition. La démarche complète est :



Il suffisait d'y penser !

25

Comment décerner un nom à un disque

Problème

A force de multiplier le nombre des disquettes, on risque de s'y perdre. Attribuer un nom à chaque disquette ou disque constituera une bonne précaution supplémentaire.

Solution

DECERNER UN NOM A CHAQUE DISQUE

Le DOS a réservé la possibilité de décerner un nom à chaque disque :

1) – Soit lors de son formatage.
2) – Soit en cours de service.

1. AU FORMATAGE

FORMAT

En lançant la commande de formatage d'une disquette, par exemple, on associera à la commande **FORMAT** une « clé » qui est **" /V "** (sans les guillemets), pour nommer le « volume ». Cela donnerait à l'écran, pour une disquette 360 Ko :

```
C:\> FORMAT A:  /V
Placez une nouvelle disquette
dans l'unité A,
et appuyez sur ENTREE

Formatage terminé

Donnez label de volume (11 car.) sinon ENTREE Clients

    362496 octets espace total
           sur disque
    362496 octets disponibles
           sur disque

Autre formatage (O/N) ?N
C:\>
```

La clé pour nommer le «volume» (le disque)

Le nom attribué à la disquette

Dès lors, on peut interroger la disquette afin de connaître son « nom de volume » avec **VOL** , ici toujours pour **A:**

```
C:\> VOL A:

 Volume dans unité A est CLIENTS

C:\>
```

De plus, l'affichage de chaque répertoire de cette disquette rappellera son nom avec, immédiatement :

```
C:\> DIR A:

 Volume dans unité A est CLIENTS
 Répertoire de  A:\

Fichier introuvable

C:\>
```

Remarquez que le DOS a fait passer le nom en majuscules. *Attention* : le nom ne doit pas comporter d'espaces ni de caractères interdits ; leur liste dépend de votre version du DOS : ne prenez pas de risque et utilisez les seuls caractères alphanumériques.

2. AVEC LA COMMANDE LABEL

On peut toujours nommer, ou renommer un disque ou une disquette en cours d'usage avec la commande **LABEL** . Celle-ci commence par ré-afficher l'ancien nom, s'il en existe un ; sinon, elle spécifiera que le disque n'avait pas été baptisé. Voici ce que cela donne, pour une disquette dans l'unité **A:** qu'on va nommer **STOCKS:**

```
C:\> LABEL A:

Le volume de l'unité A n'a pas de label

Donnez label de volume
(11 car.) sinon ENTREE
Stocks


C:\>
```

Si la disquette avait déjà été nommé, **CLIENTS** , par exemple, on aurait obtenu (en la renommant **VENTES** , ici) :

```
C:\> LABEL A:

Le volume de l'unité A est CLIENTS

Donnez label de volume
(11 car.) sinon ENTREE
VENTES


C:\>
```

A nouveau, chaque **VOL** ou chaque **DIR** réaffichera le nom du disque, ce qui sera parfois bien utile ; par exemple, lorsqu'on dispose de plusieurs disquettes **CLIENTS** répartis par ordre alphabétique. Dans ce cas, on les appellera **CLIENTSA** et **CLIENTSM** , par exemple.

26

L'exploration des fichiers d'un disque

Problème

Qu'y a-t-il réellement comme fichiers, sur un disque ? Un **DIR** *ne fournit que les noms des fichiers situés dans le répertoire actif ; mais si l'on dispose d'une hiérarchie de sous-répertoires, comment lister également ce qu'ils contiennent ?*

Solution

CHKDSK

La commande de test de disques, du DOS qui s'écrit **CHKDSK** , pour « *Check Disk* », dispose d'une *clé* fort commode, **/V** . Si l'on commande un :

```
C:\> CHKDSK /V
```

l'écran va lister tous les noms des fichiers se trouvant sur le disque visé, qu'ils proviennent de répertoires ou de sous-répertoires, en les spécifiant, d'ailleurs. Voici un extrait de ce que l'auteur a obtenu :

```
        C:\WINDOWS\TERMINAL.EXE
        C:\WINDOWS\ABC.TXT
        C:\WINDOWS\SEMAINE.TXT
        C:\WINDOWS\WRITE.EXE
        C:\WINDOWS\EXERCICE.WRI
Répertoire C:\WINDOWS\PIF
        C:\WINDOWS\PIF\PIFEDIT.EXE
        C:\WINDOWS\PIF\123.PIF
        C:\WINDOWS\PIF\123-2.PIF
        C:\WINDOWS\PIF\BASIC.PIF
        C:\WINDOWS\PIF\BASICA.PIF
        C:\WINDOWS\PIF\CHART.PIF
```

On trouve là un sous-répertoire, WINDOWS, et un sous-répertoire, PIF.

On pourrait enregistrer tout cela dans un nouveau fichier, appelé FICHIERS, par exemple, avec une *redirection* (symbole **>**) ; plus rien n'apparaîtra alors à l'écran :

```
C:\> CHKDSK /V > FICHIERS

C:\>
```

Par contre, FICHIERS est créé et contient la liste de tous les noms de fichiers. Si, au lieu de *rediriger* la sortie sur FICHIERS on l'avait redirigée vers l'imprimante, avec :

```
C:\> CHKDSK /V > PRN
```

celle-ci aurait listé ces noms de fichiers avec, pour commencer et dans le cas de l'auteur :

```
Volume LILEN
créé le 4 Mai 1987 12:43
Répertoire C:\
      C:\IBMBIO.COM
      C:\IBMDOS.COM
      C:\COMMAND.COM
Répertoire C:\VENTE
      C:\VENTE\LH2.EXE
      C:\VENTE\H1.BAS
      C:\VENTE\H2.BAS
      C:\VENTE\H3.BAS
      C:\VENTE\BOTTIN.BAT
```

Accessoirement, songez à utiliser **CHKDSK** dans le but de vérifier un disque (son rôle essentiel), et, éventuellement, de corriger certaines erreurs qu'il aurait pu détecter, en utilisant son autre clé, **/F** .

27

Ne reformatez pas accidentellement votre disque dur

Problème

On veut formater une disquette, en A, et l'on frappe **FORMAT** *à partir de C, en oubliant de spécifier A. Votre version du DOS vous a peut-être demandé une confirmation mais, sûr de vous et un peu distrait, vous avez persisté et signé. Et vous avez perdu tout ce que contenait le disque dur ! Voici comment éviter cet accident stupide.*

Solution

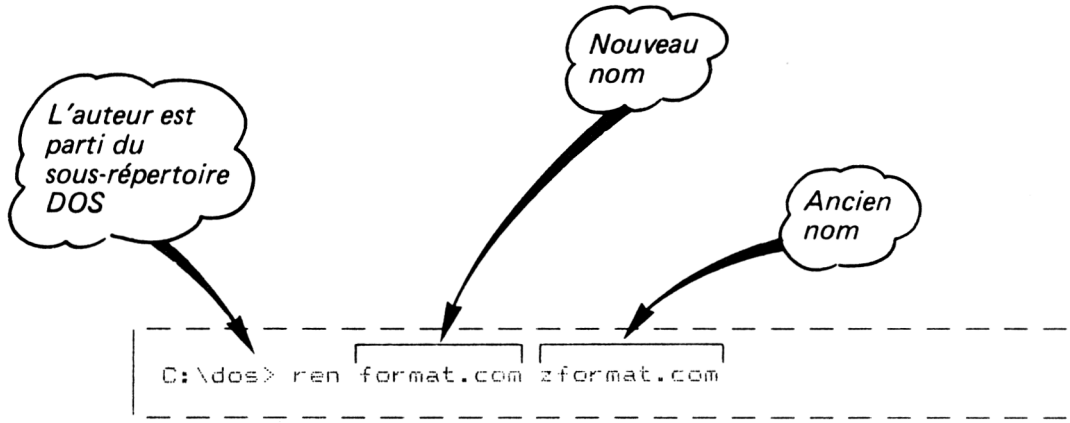
On a souvent besoin de formater des disquettes, introduites une à une dans l'unité A. Hélas, il suffit d'un instant d'inattention ou de distraction pour reformater, à la place, le disque dur C. Tout ce qu'il contenait est alors perdu !

ZFORMAT

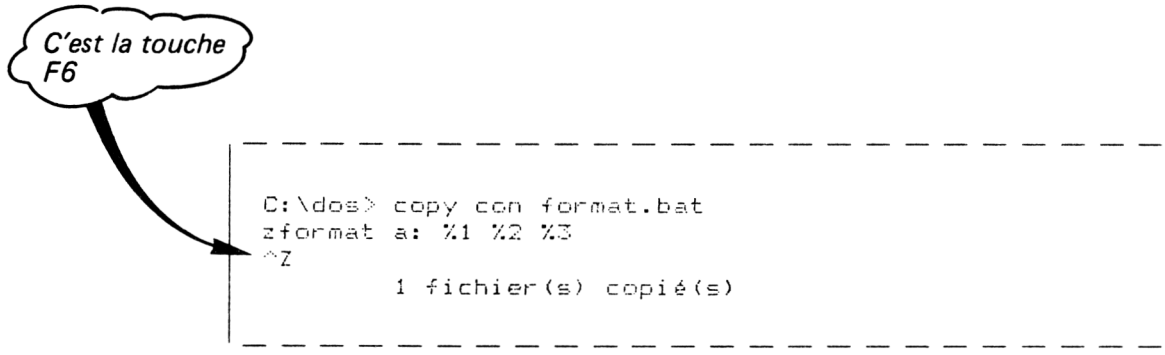
Or, il existe un moyen très simple pour exclure totalement ce genre d'incident ; il consiste à spécialiser la commande **FORMAT** à la seule unité à disquettes, en A, et à créer un nom complexe, par exemple **ZFORMAT** (ou SOS, ou...) pour le programme qui pourra reformater le disque dur. Voici le processus.

FORMAT

En fait, et parce que d'origine, c'est un seul et unique programme qui exécute tout cela, **FORMAT**, on commence par le renommer :

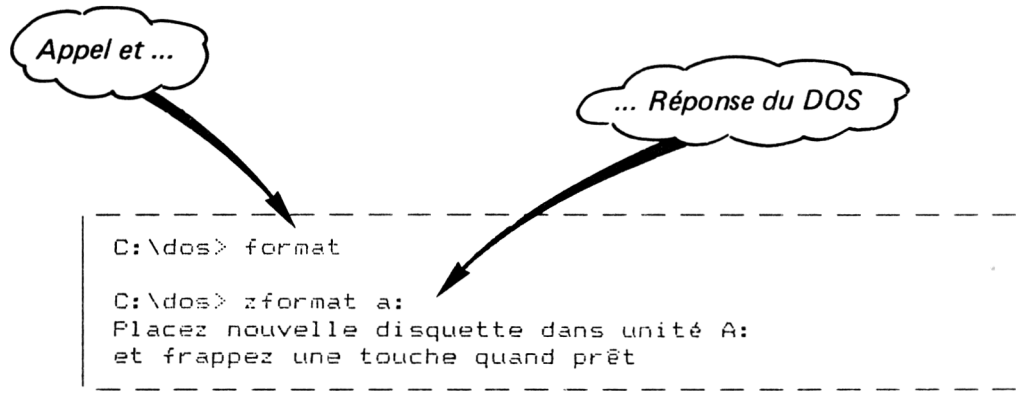


Puis, on crée un fichier **.BAT** portant *l'ancien nom*, donc **FORMAT** , devenu ici **FORMAT.BAT.** Il est tellement simple qu'on va le créer sous un **COPY CON** . Ecrivez bien la ligne avec les **%1** , **%2** et **%3** dont on va voir la signification :



La première ligne titre le fichier. La seconde représente la seule et unique commande de ce fichier : l'appel de **ZFORMAT** , *mais s'appliquant à l'unité A* obligatoirement ! Les **%1** , puis **%2** et **%3** qui suivent sont des codes non secrets du DOS : ils remplacent les indications que vous pourriez associer à **FORMAT** . Par exemple, l'auteur aime bien décerner un nom à chaque disquette : ça ne mange pas de pain et ça peut aider. Pour cela, la commande est **FORMAT /V** . Ce **" /V "** s'appelle un *paramètre* et il est symbolisé par **%1** ; grâce à **%2** et **%3** , on pourrait placer encore deux autres paramètres, ce qui est généralement totalement superflu. Le « Contrôle Z » final, noté **^Z** sur l'écran, termine ce fichier et déclenche son enregistrement (c'est la touche F6 qui l'affiche).

Désormais, si vous frappez **FORMAT** , c'est ce fichier de commandes qui va se déclencher et vous lirez à l'écran :



Avec des clés supplémentaires, par exemple **/S** pour transférer le système et **/V** pour titrer le disque, on obtiendrait, pour une séquence complète :

```
C:\dos> format /v/s

C:\dos> zformat a: /v/s
Placez nouvelle disquette dans unité A:
et frappez une touche quand prêt

Formatage en cours...Formatage terminé
Système transféré

Label de vol.(11 car. ENTREE si aucun)? Essai86

   362496 octets espace total sur disque
    40960 octets utilisés par le système
   321536 octets disponibles sur disque

Autre formatage (O/N)?n
C:\dos>
```

Enfin, on peut toujours formater le disque dur si on le désire, mais en appelant *directement* **ZFORMAT** :

```
C:\> ZFORMAT
Frappez une touche pour formater C:  ^C
```

Ici, on est reparti du répertoire principal mais le **PATH** déclaré antérieurement, ramène à **/DOS** . Ne vous privez donc pas d'une telle sécurité.

NB. — Un reformatage accidentel du disque dur peut encore être réparé si l'on possède l'un de ces programmes utilitaires, tels que *DS Recover, Mace Utilites* ou *The Norton Utilities Advanced Edition,* qui disposent de la commande UNFORMAT. Mais il vaut mieux prévenir que guérir.

28

Comment remplacer une unité à disque par une autre

Problème

On a rédigé un programme qui fait nommément appel au disque C. Or, on veut le faire tourner sur un système ne comportant que des unités à disquettes A et B. Peut-on, sans ré-écrire ce programme, détourner sur B tout ce qui était prévu pour C au préalable ?

Solution

ASSIGN

Il s'agit là d'un problème pour utilisateurs très avertis ! Il existe, en effet, une commande *externe* du DOS qui permet de substituer une unité à disque à une autre ; c'est **ASSIGN** (assignation). Sous DOS, on déclarera, par exemple :

```
ASSIGN C = B
```

Dès lors, le système s'adressera à B chaque fois qu'on lui demandera de travailler avec C.

Attention, car désormais, le système va ignorer le disque C, quand bien même il existerait !

Pour annuler une assignation, on frappe simplement :

```
ASSIGN
```

Ou encore, on ré-initialise le système. On pourrait encore, sur une seule ligne **ASSIGN** , multiplier les ré-affectations et écrire, dans un tout autre cas, par exemple.

```
ASSIGN A=C  B=C
```

Dans ce cas, tous les appels à A ou B seraient dirigés sur C.

En utilisant **ASSIGN** , il convient d'être prudent et de bien se souvenir des réassignations. D'ailleurs, certaines commandes les ignoreront (**FORMAT, DISCOPY** et **DISKCOMP**), alors que d'autre ne devront absolument pas être utilisées : **BACKUP, RESTORE, LABEL, JOIN, SUBST** ou **PRINT** .

29

Comment récupérer un fichier effacé par accident

RECUPERER UN FICHIER EFFACE

Problème

*On a effacé, par erreur, un fichier précieux sur disque. Hélas, on n'a pas cru utile d'en établir une copie de sécurité au préalable. Cruelle négligence, qui, maintenant, vous coûte cher (à l'inverse des assurances : elles prétendent ne coûter cher qu'**avant** un sinistre). Peut-on récupérer ce qu'on a malencontreusement effacé ? Oui !*

Solution

L'effacement accidentel d'un fichier est, hélas, un accident courant. Encore heureux quand il ne s'agit pas du contenu complet d'un répertoire ! Méfiez-vous des `DEL` ou `ERASE` , surtout en fin de journée, quand l'attention n'est plus aussi soutenue. On confirme alors trop vite et on détruit ce qu'on avait de plus précieux. Le reconstituer pourrait demander des jours ou des semaines de travail, si, encore, c'est possible.

Si vous avez suivi les conseils qu'on vous prodigue généreusement, vous aurez établi des copies de sécurité et vous pourrez rétablir votre fichier. Mais si tel n'est pas le cas (honte sur vous ; mais soyez persuadé que personne ne pourra certainement vous jeter la première pierre), il vous reste encore un fort espoir : appelez à la rescousse l'un des nombreux programmes de restitution de fichiers. Ils sont vendus séparément ; voici les principaux :

- *The Norton Utilities ;*
- *Mace Utilities ;*
- *PC Tools ;*
- *Power Tools ;* etc.

L'auteur utilise couramment les *Norton* (ils existent en plusieurs versions) et *PC Tools*. Mais pourquoi une telle récupération est-elle possible ?

En réalité, et lorsqu'on efface un fichier, le DOS ne l'efface pas réellement : il subsiste, intact, sur le disque. Le DOS perd toutefois connaissance de son existence. Pour lui, c'est comme s'il n'existait plus. Il subsistera sur le disque jusqu'à ce qu'on enregistre un autre fichier sur l'espace qu'il occupait. Aussi, tenez bien compte de cet avertissement :

Attention : après l'effacement accidentel, ne faites plus rien sur machine, et surtout, n'enregistrez absolument rien sur le disque victime de l'accident.

Passez à la séquence de récupération ! Mais examinons plus en détail ce qui se passe. Sur commande d'effacement, le DOS va marquer d'un caractère spécial le nom du fichier dans le répertoire ; il modifie, en fait, le premier caractère du nom du fichier et le remplace par le caractère ASCII dont le code, en hexadécimal est `E5` . En outre, il remet à jour la table spécifique qui lui indique où se trouvait le fichier, supprimant les indications qui y étaient relatives et les remplaçant par des marques de disponibilité. Cette table s'appelle la FAT, de l'anglais *« File Allocation Table »*, *Table d'affectations des fichiers.*

Par conséquent, la reconstitution d'un fichier va demander :

1) — Qu'on rétablisse le nom du fichier, en remplaçant le caractère `E5` par la lettre de l'alphabet manquante.

2) — Que les « entrées » dans la FAT soient reconstituées.

Vous seul pouvez indiquer au programme de restitution quelle lettre il faut affecter au nom du fichier. Par exemple, et avec `?ENTES.MAI` , ce sera un `V` ! Le programme utilitaire, lui, se chargera de retrouver à quels emplacements se trouvait votre fichier effacé ; c'est simple, le plus souvent, il lui suffira de suivre la piste des positions disponibles sur le disque, dans le bon ordre. Pour les curieux : les fichiers sont rangés par blocs appelés des *« clusters »*, en franglais ; la longueur d'un cluster dépend du disque et du DOS ; elle pourrait être, par exemple, de deux secteurs de disque, ou davantage. L'utilitaire va le vérifier, puis il relèvera les « trous », la présence de clusters devenus disponibles sur le disque, au milieu d'autres fichiers complets, et les utilisera pour rétablir le vôtre.

CLUSTERS

Avec un seul fichier effacé, et aucune opération entre temps, les chances de réussite sont très élevées.

En effet, ça peut ne pas fonctionner à tous les coups. Dans la pratique, la reconstitution sera plus délicate lorsque plusieurs fichiers auront été simultanément effacés. C'est pourquoi l'utilitaire offre souvent un mode *automatique*, à appliquer lorsqu'il vous signifiera que tout lui paraît facile, et un mode *manuel* dans les autres cas.

NORTON UTILITIES

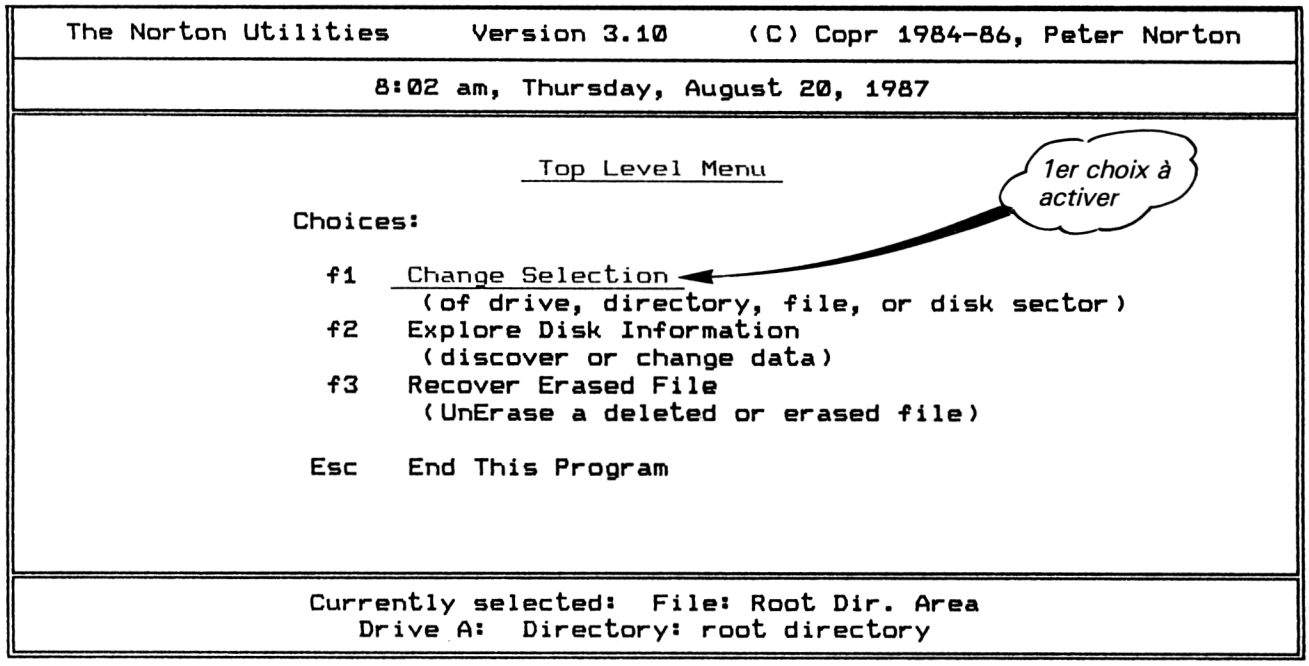
Voici comment, avec les *Norton Utilities*, on procède pour récupérer un fichier `VENTES.MAI` effacé par erreur. Tout d'abord, voici ce qu'il ne fallait absolument pas faire :

```
C:\VENTE> DEL VENTES.MAI

C:\VENTE>
```

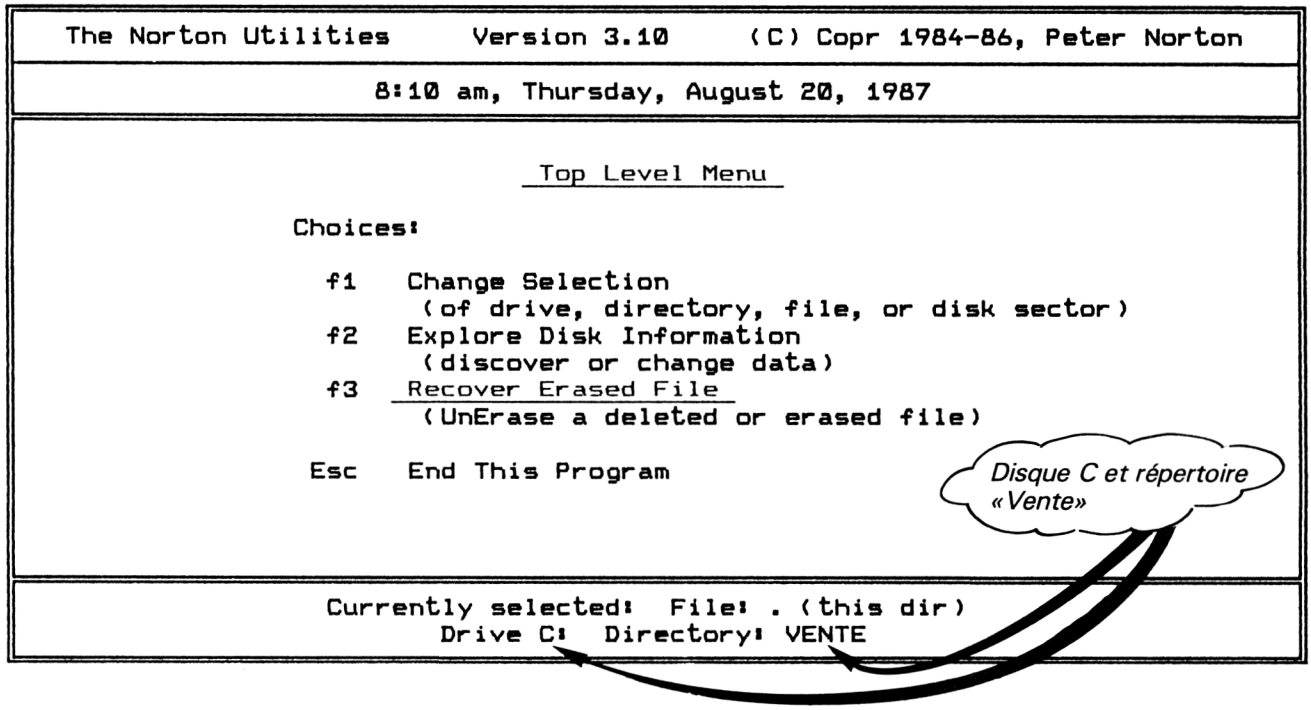
La disquette de l'utilitaire est insérée en A et on appelle ce programme :

```
C:\VENTE> A:

A:\> NORTON
```

Dès lors, il va suffir de se laisser guider par les menus successifs, en évitant même de prendre des initiatives. Voici le film de ce qui va se passer, avec le minimum de commentaires. Le premier écran de *Norton* est son *menu* principal (le plus difficile, si l'anglais, c'est pour vous de l'hébreu, sera d'en comprendre la signification) :



Il faut, en effet, indiquer à l'utilitaire qu'on travaille avec le disque C et le répertoire VENTE. La frappe sur *F1* ou sur *Entrée* permet de fournir ces informations. Dès lors, on passe à *F3, « Récupérer le fichier effacé ».*



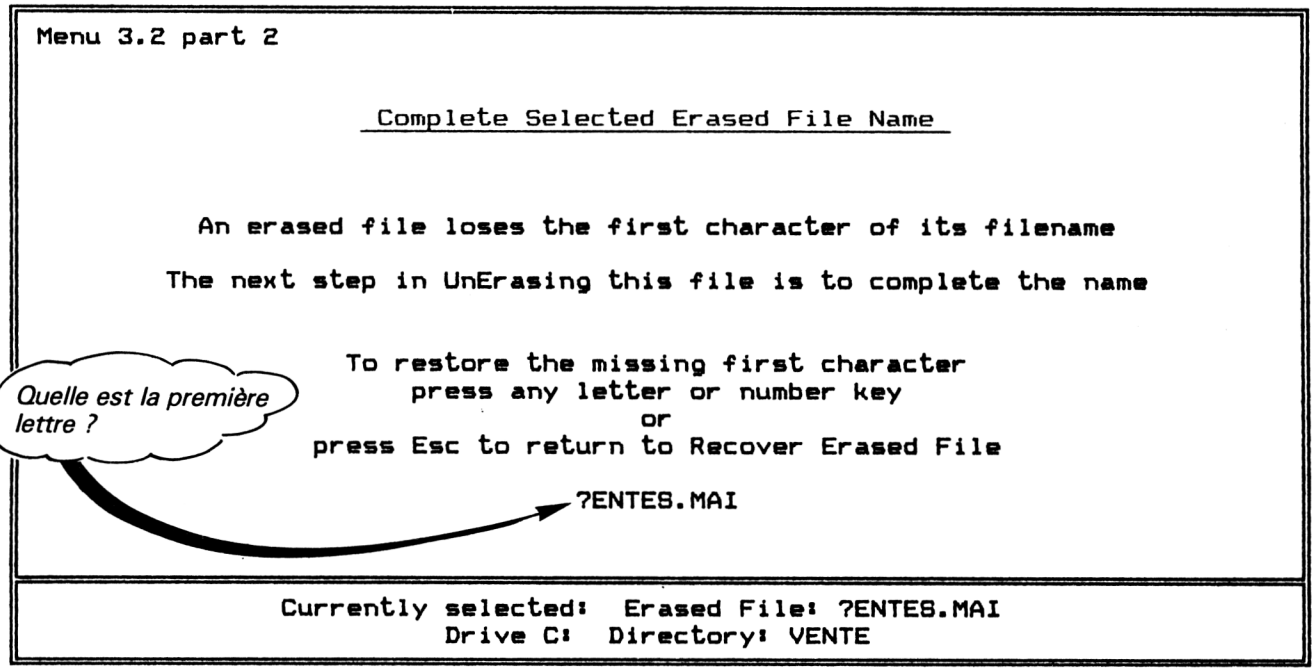
Le choix *F3* fait passer dans un sous-menu où l'on va sélectionner *« F2 Choisir un fichier effacé à reconstituer » :*

```
Menu 3
                          Recover Erased File

            Choices:

              f1   Change Selected Drive and Directory

              f2   Select an Erased File to UnErase

              f3   Display Information about Erased File

              f4   Find Erased File's Data

              f5   Save UnErased File

             Esc   Return to Top Level Menu

   See the Norton Utilities manual for an outline of the steps to follow

                 Currently selected:  File: . (this dir)
                        Drive C:  Directory: VENTE
```

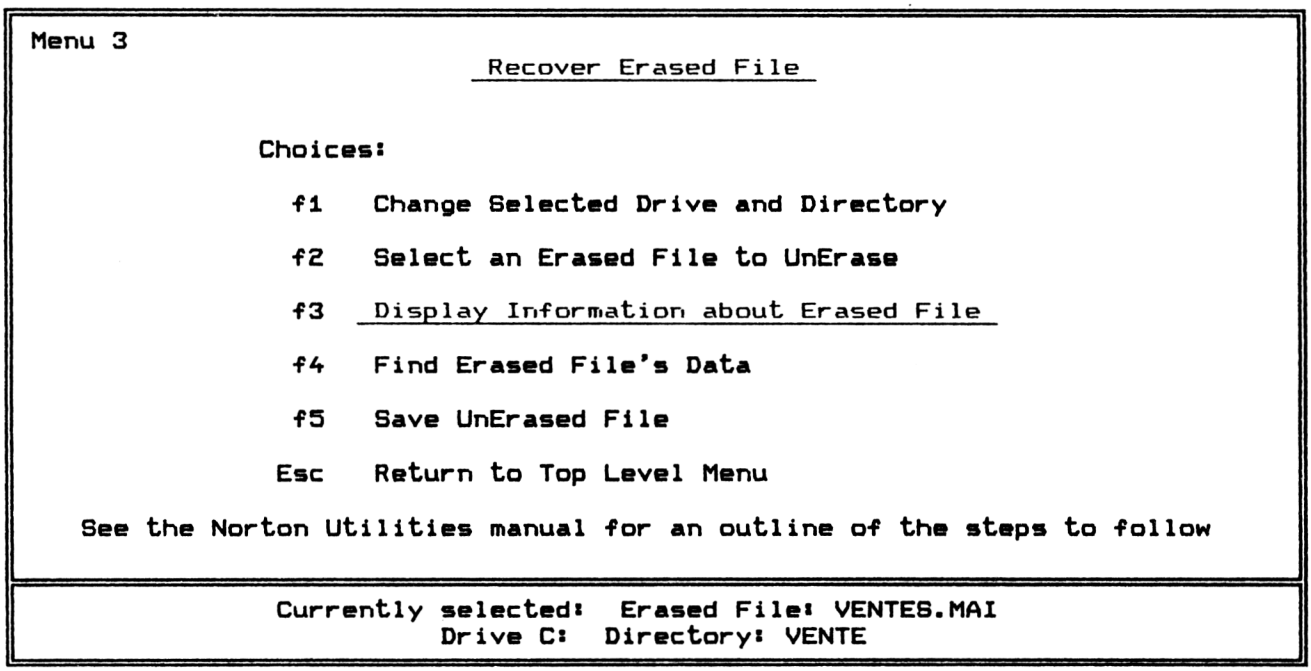
Aussitôt, l'utilitaire explore le disque et fournit le nom amputé de tous les fichiers effacés qu'il a détectés ; ici, il n'y en a heureusement qu'un :

```
Menu 3.2 part 1
                          Select Erased File

                              ?ENTES.MAI




  1 to choose from;  use cursor keys;  press Enter to take selection
                Select another disk or directory to see others

                 Currently selected:  Erased File: ?ENTES.MAI
                        Drive C:  Directory: VENTE
```

On le sélectionne donc ; l'utilitaire va nous demander sa première lettre :



Dès qu'on frappe un V, le « Menu 3 » revient et suggère de passer à *« F3 Afficher des informations relatives au fichier effacé » :*



Oui ! L'utilitaire va alors offrir l'écran suivant :

```
Menu 3.3

                 Display Information about an Erased File

                         Name: VENTES.MAI
                   Attributes: Archive
                Date and time: Wednesday, July 30, 1986, 1:51 am
       Starting cluster number: 2,246 (sector number 18,001)
                         Size: 28,346 bytes, occupying 7 clusters

          The beginning of the space formerly used by this file
          is not in use by the data of another file.

          Successful UnErasure of this erased file is possible.

            Press any key to continue...
```

«La récupération de ce fichier est parfaitement faisable !»

Avec l'*Entrée* suivante, le même *Menu 3* propose de passer à *« F4 Rechercher les données du fichier effacé » :*

```
Menu 3
                          Recover Erased File

              Choices:

                f1   Change Selected Drive and Directory

                f2   Select an Erased File to UnErase

                f3   Display Information about Erased File

                f4   Find Erased File's Data

                f5   Save UnErased File

               Esc   Return to Top Level Menu

   See the Norton Utilities manual for an outline of the steps to follow

              Currently selected:  Erased File: VENTES.MAI
                        Drive C:  Directory: VENTE
```

Si l'on confirme, un sous-menu du *Menu 3*, appelé *Menu 3.4*, suggère de récupérer *automatiquement* tous les *clusters* du fichier effacé, en faisant confiance à M. Norton :

```
Menu 3.4
                        Find Erased File's Data

        File data is stored in units of disk space called clusters

            The erased file VENTES.MAI occupied 7 clusters


            Choices:

              f1   Automatically Select All Data Clusters (Best Guess)
              f2   Automatically Select One Cluster (Best Guess)
              f3   Search for Specific Data
              f4   Select One Cluster by Sector Number
             Esc   Return to Recover Erased File

              Currently selected:  Erased File: VENTES.MAI
                                   Drive C:  Directory: VENTE
```

L'opération démarre sur la frappe de F1, ou de *Entrée* ; désormais, l'utilitaire parle à l'imparfait du fichier « effacé », car il va le récupérer si l'on choisit *« Esc Retourner pour récupérer le fichier effacé » :*

```
Menu 3.4
                        Find Erased File's Data

        File data is stored in units of disk space called clusters

            The erased file VENTES.MAI occupied 7 clusters
            You have selected 7 clusters

            To finish UnErasing the file, go to menu 3.5
            You may review the selected data at menu 3.3
            To undo the selection so for, go to menu 3.2

            Choices:

              f1   Automatically Select All Data Clusters (Best Guess)
              f2   Automatically Select One Cluster (Best Guess)
              f3   Search for Specific Data
              f4   Select One Cluster by Sector Number
             Esc   Return to Recover Erased File

              Currently selected:  Erased File: VENTES.MAI
                                   Drive C:  Directory: VENTE
```

C'est chose faite ! Maintenant, l'utilitaire indique qu'il vaut mieux sauvegarder le fichier sur disque, avec *« F5 Sauvegarder le fichier récupéré »* :

```
Menu 3

                    Recover Erased File

          Choices:

            f1   Change Selected Drive and Directory

            f2   Select an Erased File to UnErase

            f3   Display Information about Erased File

            f4   Find Erased File's Data

            f5   Save UnErased File

           Esc   Return to Top Level Menu

   See the Norton Utilities manual for an outline of the steps to follow

          Currently selected:  Erased File: VENTES.MAI
                    Drive C:   Directory: VENTE
```

Après le choix de F5, cette dernière opération s'exécute ; voici le bulletin de victoire :

```
Menu 3.5

                    Save UnErased File


          Saving record of file's data ... done

          Saving file's directory entry ... done


          UnErasing VENTES.MAI is complete

          Press any key to continue...


          Currently selected:  File: VENTES.MAI
                    Drive C:   Directory: VENTE
```

Toute l'opération n'aura pris qu'une *dizaine de secondes*, pour quelqu'un ayant l'expérience de cet utilitaire. Pourquoi ne pas vous entraîner, avec quelques exercices et à temps perdu, avant tout accident ?

Dès lors, le fichier est, à nouveau, disponible et complet. Le prix de cet utilitaire (il a beaucoup varié dans le temps) est très largement couvert pas tout ce qu'il vous aura épargné.

30

Comment afficher un document par écrans complets ?

TYPE

Problème

Lorsqu'on veut afficher un document qui occupe plus d'un écran (un répertoire, par exemple, ou encore, et avec un `TYPE`*, un document ASCII), celui-ci se met à défiler sans qu'on ait le temps de le lire. Comment pallier ce phénomène ?*

Solution

CONTROLE + VERROUILLAGE MAJUSCULES

MORE

Deux méthodes permettent de bloquer un texte à l'écran :

· La frappe de *Contrôle* + *Verrouillage majuscules* (du pavé numérique) stoppe un défilement. La frappe d'une touche quelconque permet, ensuite, de le reprendre. Cette méthode demande une certaine dextérité manuelle, un bon coup d'œil et des réflexes de sportif.
· Une seconde méthode, plus rationnelle, consiste à recourir à la commande `MORE` du DOS.

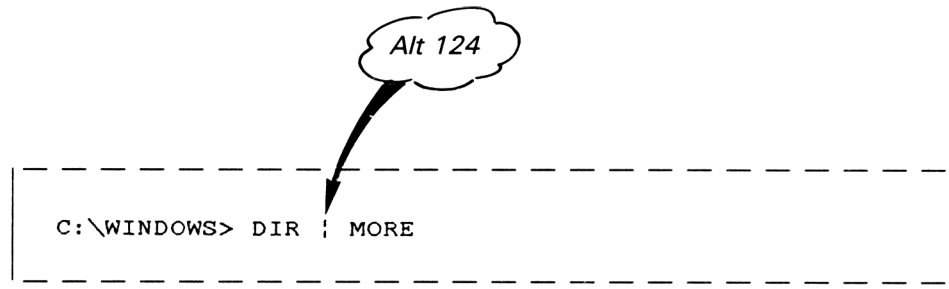
MORE provoque un affichage sur un plein écran. En bas, il inscrit le mot « *Suite* ». La frappe d'une touche quelconque affichera l'écran suivant, toujours avec « Suite ». Et ainsi, jusqu'à la fin du document.

Notez que **MORE** est un programme *externe* du DOS. Cela signifie que le système doit pouvoir l'atteindre, dans quelque répertoire qu'il se trouve, lorsqu'il est invoqué.

En voici un premier exemple. Le répertoire de WINDOWS ne tient pas sur un écran. On va donc chaîner la commande DIR avec un **MORE** .

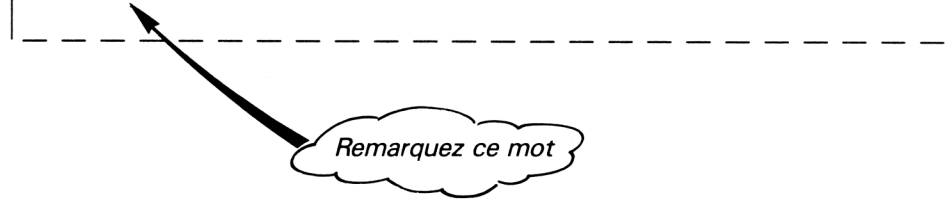
Pour chaîner des commandes, on place le symbole « trait vertical interrompu ». Il s'obtient en maintenant la touche *Alt* enfoncée et en frappant le nombre 124 (code ASCII de ce symbole) sur le clavier numérique exclusivement.

N'essayez pas les touches numériques du clavier alphabétique, cela ne fonctionnerait pas. Voici la commande :



Et le résultat :

```
 Volume dans unité C est LILEN
 Répertoire de  C:\WINDOWS

.             <DIR>       1/01/80    2:25
..            <DIR>       1/01/80    2:25
WIN100   OVL   215968    3/12/86   14:01
WIN      INI     2973   23/04/87   15:27
SPOOLER  EXE    13392    3/12/86   14:02
WINOLDAP MOD    20176    3/12/86   14:02
WINOLDAP GRB     1374    3/12/86   14:02
MSDOS    EXE        1   23/04/87   14:50
WIN      COM     4969    3/12/86   14:01
WIN100   BIN   181744    3/12/86   14:01
IBMGRX   DRV    12288   11/01/87   14:30
COURA    FON     8720    3/12/86   14:03
HELVA    FON    27056    3/12/86   14:03
TMSRA    FON    26256    3/12/86   14:03
COURD    FON    15136    3/12/86   14:03
HELVD    FON    45600    3/12/86   14:03
TMSRD    FON    45200    3/12/86   14:03
COURC    FON     8784    3/12/86   14:03
HELVC    FON    26624    3/12/86   14:03
TMSRC    FON    25824    3/12/86   14:03
- Suite -
```



La frappe d'une touche quelconque affichera l'écran suivant, et ainsi de suite jusqu'à la fin.

Second exemple, on dispose d'un long document en ASCII à afficher. Ce pourrait être un texte du genre de ceux qui accompagnent certains logiciels pour nous fournir des précisions de dernière heure ; ils s'appellent souvent **LISEZ.MOI** , ou **LISEZ.SVP** , ou **LISEZ.DOC** , etc. Ici, nous allons appeler un document dont le nom est **PAO.ASC** d'abord selon la méthode « longue » de chaînage des commandes, avec un **TYPE** pour afficher, puis notre **MORE** :

```
C:\PAO> TYPE PAO.ASC ¦ MORE
```

Voici le premier écran de ce document :

```
                    Pratique de la publication assistée par ordinateur --  PAO
---

                        Chapitre  I

          Introduction  à  la PAO



Publication assistée par ordinateur (PAO en abrégé), édition assistée, micro-édi
tion sont des termes synonymes traduisant ce que les Américains appellent  le "d
esktop publishing".  Dans ce premier chapitre,  on va en étudier les caractéris
tiques les plus générales et les applications, souvent sous forme de synthèse. P
ar  la suite, on reviendra plus en détail sur tel ou tel point. On va également
trouver ici aussi bien de premières recommandations et conseils que quelques mis
es en garde.

- Suite -
```

A nouveau, le mot «Suite» indique que d'autres écrans suivent

La frappe d'une touche quelconque provoque l'affichage de la suite, d'un second écran, etc. Mais on peut simplifier la commande en utilisant le symbole de *redirection*, ici le **"<"** ; la commande suivante signifie : *diriger* **PAO.ASC** *vers* **MORE** *pour l'afficher :*

La redirection

```
C:\PAO> MORE < PAO.ASC
```

Le résultat sera exactement le même. *Attention :* ne vous trompez toutefois pas de symbole et ne frappez pas un **>** au lieu d'un **<** , car dans ce cas, c'est le **MORE** qui serait redirigé vers **PAO.ASC** . Comme il n'a rien à écrire, il effacerait le fichier !

31

Comment afficher en surbrillance ?

Problème

On veut commander des affichages en surbrillance, sous DOS, afin de bien attirer l'attention.

Solution

ANSI.SYS

Pour cela, il faut faire appel à la séquence **ANSI.SYS** du DOS. Incluez-la dans le **CONFIG.SYS** avec une ligne.

```
DEVICE = ANSI.SYS
```

Rappelons que d'autres fiches de ce livre vous expliquent comment ajouter une ligne à un fichier tel que **CONFIG.SYS** ou **AUTOEXEC.BAT** . Copiez le fichier **ANSI.SYS** dans le répertoire principal, puis, ré-initialisez le micro afin que le système charge **ANSI.SYS** . Désormais, toutes ses commandes portant sur l'écran (gestion du curseur, effacement, etc) et le clavier sont disponibles. On va les expérimenter en affichant, en surbrillance, les caractères d'appel du DOS, vos commandes apparaissant dans la luminosité normale. Pour cela, on fait appel à **PROMPT** , qui définit ces caractères d'appel ; notre séquence va comporter :

PROMPT

- La commande **PROMPT**
- L'appel de **ANSI.SYS** , qui s'écrit **$e[**

$E[

ESC [

Sous DOS, l'écriture de " $e[" correspond à ce qui est défini par les symboles « ESC [» dans les fonctions ANSI.

- La mise en service de la surbrillance, qui s'écrit " 1m " .
- Les codes des caractères d'appel, avec le sous-répertoire en service, donc **PG** .
- Un ré-appel de **ANSI.SYS** avec le même " **$e[** " .
- Le code de fin de surbrillance, qui s'écrit " 0m " .
- Pour fignoler, on ajoutera même un **$A** , qui ne signifie rien, mais sera remplacé par un espace. Par conséquent, la frappe des commandes sera éloignée d'un caractère de la fin des caractères d'appel.

PG

Voici ce que cela donne, avec le résultat (même s'il n'est pas évident sur cette impression !) :

```
C:\> PROMPT $e[1m$p$g$e[0m $A
C:\>
```

Passons à un sous-répertoire, afin de le confirmer :

```
C:\> cd windows
C:\WINDOWS> La commande est décalée et en "normal".
```

Voici la liste des commandes disponibles sous ANSI.SYS, en 3 tableaux et en synthèse ; rappelez-vous que ESC se traduit paar **" $e "** , sous DOS, fournissant le *caractère d'échappement* 27, en code ASCII.

1. CURSEUR ET EFFACEMENT

Nom	Commande	Fonction	Commentaires
PCU	Esc[rangée;colonne*h*	Position du curseur	Déplace le curseur à la position spécifiée (par défaut à 1 : 1)
CVH	Esc[nombredelignes*a*	Déplacement du curseur vers le haut	Sans changer de colonne
CVB	Esc[nombredelignes*b*	Déplacement du curseur vers le bas	Sans changer de colonne
CVD	Esc[nombredecolonnes*c*	Déplacement du curseur vers la droite	Sans changer de ligne
CVG	Esc[nombredecolonnes*d*	Déplacement du curseur vers la gauche	Sans changer de ligne
PHV	Esc[rangée;colonne*f*	Position horizontale et verticale du curseur	Déplace le curseur (par défaut, à 1 : 1)
DPC	Esc[6*n*	Demande la position du curseur	Le contrôleur renvoie le RDC
EPC	Esc[rangée;colonne*r*	Etat de la position du curseur	Position du curseur avec périphérique d'entrée standard
MPC	Esc[*s*	Mise en mémoire de la position du curseur	On la retrouve avec la fonction suivante RDC
RDC	Esc[*u*	Repositionnement du curseur	Avec la fonction DPC
EE	Esc[2*J*	Effacement de l'écran	Efface l'écran et place le curseur à sa position initiale
EL	Esc[*k*	Effacement dans une ligne	Efface les caractères du curseur jusqu'à la fin de la ligne

2. ECRAN

Nom	Commande	Fonction
SCG	Esc[paramètre;... ;paramètre*m*	Spécifie l'attribut avec « paramètre » : Ø = Aucun attribut (blanc sur noir) 1 = Gras (surbrillance) 4 = Soulignement (monochrome) 5 = Clignotement 7 = Vidéo inverse 8 = Caractères invisibles (non affichés) 3Ø = Premier plan noir 31 = Premier plan rouge 32 = Premier plan vert 33 = Premier plan jaune 34 = Premier plan bleu 35 = Premier plan magenta 36 = Premier plan cyan 37 = Premier plan blanc 4Ø = Arrière-plan noir 41 = Arrière-plan rouge 42 = Arrière-plan vert 43 = Arrière-plan jaune 44 = Arrière-plan bleu 45 = Arrière-plan magenta 46 = Arrière-plan cyan 47 = Arrière-plan blanc 48 = Indice 49 = Exposant
RTA	Esc[= paramètre*h* Esc[= *h* Esc[= Ø*h* Esc[?7*h* (au choix)	Réglage du type d'affichage ou de la largeur, selon le paramètre, avec : Paramètre Ø = 40 × 25 monochrome 1 = 4Ø × 25 couleur 2 = 8Ø × 25 monochrome 3 = 8Ø × 25 couleur 4 = 32Ø × 2ØØ couleur 5 = 32Ø × 2ØØ monochrome 6 = 64Ø × 2ØØ monochrome 7 = Retour marge automatique à la fin d'une ligne sur la suivante
ARMA	Esc[= paramètre*i* Esc[= *i* Esc[= Ø*i* Esc[?7*i* (au choix)	Annulation du retour automatique à la marge. Les paramètres sont identiques à RTA, sauf le 7 qui annule le retour automatique à la marge.

3. CLAVIER

Commande	Fonction
Esc[#;#;....#*p* Esc[''chaîne''....;*p* Esc[#;''chaîne''....# ;# ;« chaîne;#*p* ou toute autre combinaison de chaînes ou de chiffres décimaux	Le 1er code ASCII de la séquence donne le code à redéfinir. Le reste détermine la séquence déclenchée par la nouvelle définition. Si le 1er code est Ø (NUL), le 2e fournit un code ASCII étendu.

Une fiche spécifique de ce livre traite, par ailleurs, de la reprogrammation du clavier sous **ANSI.SYS** .

32

Le bon usage des caractères de remplacement * et ? du DOS

CARACTERES DE REMPLACEMENT

Problème

*Les caractères de remplacement du DOS sont le point d'interrogation (?) et l'astérisque (*). Or, ce dernier semble ne pas fonctionner lorsqu'on le place en début de mot.*

Solution

Rappelons, d'abord, l'usage de ces caractères de remplacement *dans les noms de fichiers* :

POINT D'INTERROGATION

· *Le point d'interrogation* remplace un *unique* caractère *quelconque.* Ainsi, `CLIENT?` remplacera aussi bien `CLIENTS` que `CLIENTA` , `CLIENTSB` , `CLIENT5` , etc. On peut placer un ou plusieurs points d'interrogation successifs ou non, n'importe où dans un nom de fichier ou son extension.

ASTERISQUE

· *L'astérisque* remplace un groupe d'un nombre *quelconque* de caractères quelconques, dans les noms de fichiers ou dans les extensions. Ainsi, `*.MAI` remplacera `CLIENTS.MAI` , ou `STOCKS.MAI` , etc. ; de même, `VENTES.*` remplacera `VENTES.JAN` , ou `VENTES.FEV` , etc.

On est autorisé à placer un astérisque *en fin* d'un mot ; par exemple, et pour reprendre des noms de fichiers du DOS, `BAS*.COM` représentera `BASIC.COM` et `BASICA.COM` . Par contre, l'emploi de l'astérisque *en début* de nom est interdit ; c'est cela, le piège qui surprend les débutants, et les autres !

Supposons qu'on veuille relever les « mois en R » avec un `*BRE` portant sur un fichier regroupant les noms des 12 mois. Et bien, le DOS ne va pas se borner à `SEPTEMBRE` , `OCTOBRE` , `NOVEMBRE` et `DECEMBRE` , il va les lister *tous*, à partir de `JANVIER` .

L'explication est simple : un nom de fichier peut comporter jusqu'à 8 caractères. Ces caractères étant quelconques, représentons-les par des points d'interrogation (?) ; un nom pourra donc être ???????? Or, l'astérisque remplace un nombre *quelconque*, non fixé, de caractères, donc de points d'interrogation. Par conséquent, ***BRE** représentera aussi bien :

```
?BRE
??BRE
???BRE
????BRE
?????BRE
??????BRE
???????BRE
????????BRE
```

Les caractères exédant le nombre de 8 étant perdus, la dernière ligne est équivalente à **????????** , donc à un nom quelconque. Selon les dernières lignes, ces noms pourraient aussi bien être **JANVIER** que **FEVRIER** ou **MARS** , etc. Méfiez-vous donc de telles situations.

33

Comment créer un fichier en le frappant directement au clavier

Problème

On veut créer un petit fichier de texte. Doit-on obligatoirement passer par un traitement de texte complet ou existe-t-il une méthode plus simple et plus directe ?

Solution

COPY

EDLIN

CON

Deux méthodes s'offrent à l'utilisateur : la création directe, au clavier, avec la commande `COPY` , ou une création via l'éditeur `EDLIN` . Nous débattons de ces thèmes dans diverses autres fiches, car il s'agit là d'un problème que les utilisateurs posent fréquemment. Ici, nous allons nous borner à étudier la création d'un fichier par frappe directe, au clavier, grâce à une commande très subtile, `COPY` .

Pour le système, l'ensemble clavier-écran s'appelle *la console* ; on utilise `CON` , en abrégé, éventuellement suivi d'un double-point (`CON:`) ; certains DOS exigent ce double-point alors que d'autres s'en passent allègrement.
Derrière la commande `COPY` :
· Si l'on frappe un nom de fichier, puis `CON:` , cela signifie *« Copier le fichier nommé à l'écran »*, donc l'afficher.
· Si l'on frappe `CON:` suivi par un nom de fichier, cela signifie *« Enregistrer les frappes au clavier dans le fichier nommé »*.

Imaginez qu'on veuille créer un fichier appelé SALUTS et ne contenant que ces mots : *« Bonjour, Madame »*. Le processus va être le suivant :

COPY CON

CONTROLE Z

— On frappe `COPY CON: SALUT` , puis *Entrée.*
— Le système attend d'autres frappes ; il ne ramène donc pas les caractères d'appel du DOS à l'écran.
— On frappe *« Bonjour Madame »*, puis *Entrée.*
— On pourrait ajouter d'autres lignes, à volonté, mais comme on préfère s'en tenir là, on « referme » ce fichier. C'est la frappe sur la touche F6 qui marque sa fin ; elle affiche les caractères `^Z` à l'écran, pour « Contrôle Z ». On aurait pu frapper la touche *Contrôle*, la maintenir enfoncée, puis frapper un Z, mais c'eût été plus long !

– Au dernier *Entrée*, le fichier est enregistré, ce qu'indique le DOS en écrivant « 1 Fichier(s) copié(s) », et les caractères d'appel du DOS ressurgissent :

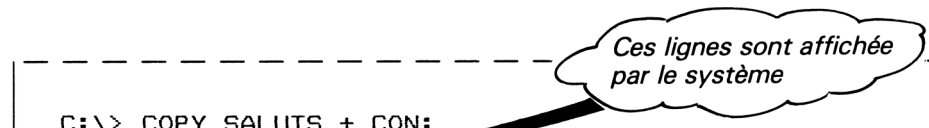
```
C:\> COPY CON: SALUTS
Bonjour, Madame
^Z
        1 Fichier(s) copié(s)

C:\>
```

Pour ré-afficher ce fichier, on fera :

```
C:\>COPY SALUTS CON:
Bonjour, Madame
        1 Fichier(s) copié(s)

C:\>
```

PRINCIPE DE LA CONCATENATION

D'autre part, on peut reprendre un fichier et lui ajouter des lignes en fin de texte. C'est ce qu'on appelle *concaténer du texte.* Le symbole d'ajout de lignes est tout simplement le signe *plus* (**+**). Ainsi, pour ajouter une ligne au fichier précédent, on fera un **COPY SALUTS + CON:** , donc, *plus* ce que l'on frappera au clavier :




```
C:\> COPY SALUTS + CON:
SALUTS
CON
Nous vous souhaitons une bonne journée.
^Z
        1 Fichier(s) copié(s)
```

De même, on peut relire ce fichier, afin de vérifier que la concaténation s'est bien opérée :

```
C:\> COPY SALUTS CON:
Bonjour, Madame
Nous vous souhaitons une bonne journée.
        1 Fichier(s) copié(s)

C:\>
```

En pratique, employez **COPY CON:** pour des fichiers tels que les **CONFIG.SYS** , **AUTOEXEC.BAT** et autres fichiers **.BAT** que vous serez amené à créer. Nous y reviendrons à plusieurs reprises dans ce livre.

34

Comment ajouter des lignes dans AUTOEXEC.BAT ou CONFIG.SYS (ou les créer) ?

Problème

Pour installer tel ou tel programme, ou pour améliorer la gestion du système, on conseille (parfois fortement) d'introduire des lignes spéciales, soit dans le fichier `AUTOEXEC.BAT` *, soit dans le fichier* `CONFIG.SYS` *. Comment faire pour écrire ces lignes si l'on est novice ?*

Solution

Trois méthodes peuvent être couramment employées pour introduire de nouvelles lignes dans les fichiers `AUTOEXEC.BAT` , `CONFIG.SYS` , ou autres :

- Utiliser la commande `COPY` du DOS.
- Passer par son « éditeur de lignes » `EDLIN` .
- Appeler un traitement de texte classique.

On va les examiner séparément.

1. AVEC COPY

COPY

CON:

Programme *interne* du DOS, `COPY` s'accommode à de multiples sauces. C'est `COPY` qu'on utilise essentiellement pour copier des fichiers d'un disque à l'autre ; mais il peut aussi intervenir pour *copier* à partir de l'écran, ou encore pour *afficher sur l'écran.* L'ensemble clavier-écran étant *la console*, pour le système, son abrégé est `CON` . Dans les ordres suivants, on fera suivre CON d'un double-point (`CON:`), bien que le plus souvent, on puisse s'en dispenser.

TYPE

Supposons qu'on veuille ajouter, dans le fichier **CONFIG.SYS** , la ligne **"BUFFER = 30"** . Il s'agit de zones tampons, en mémoire (si celle-ci est suffisante), qui accélèrent les échanges avec le disque. On commence *toujours*, par sécurité, par afficher le contenu *actuel* du fichier, avant modification, et cela, grâce à la commande **TYPE** *suivie du nom du fichier.*

```
C:\> TYPE CONFIG.SYS
COUNTRY = 33
FILES   = 20
BUFFERS = 30

C:\>
```

Ce fichier contient deux lignes, **COUNTRY = 33** et **FILES = 20** ; l'auteur avait bien aligné les nombres, luxe inutile. *Dès lors, et pour ajouter notre ligne BUFFER = 30, deux méthodes sont applicables :*

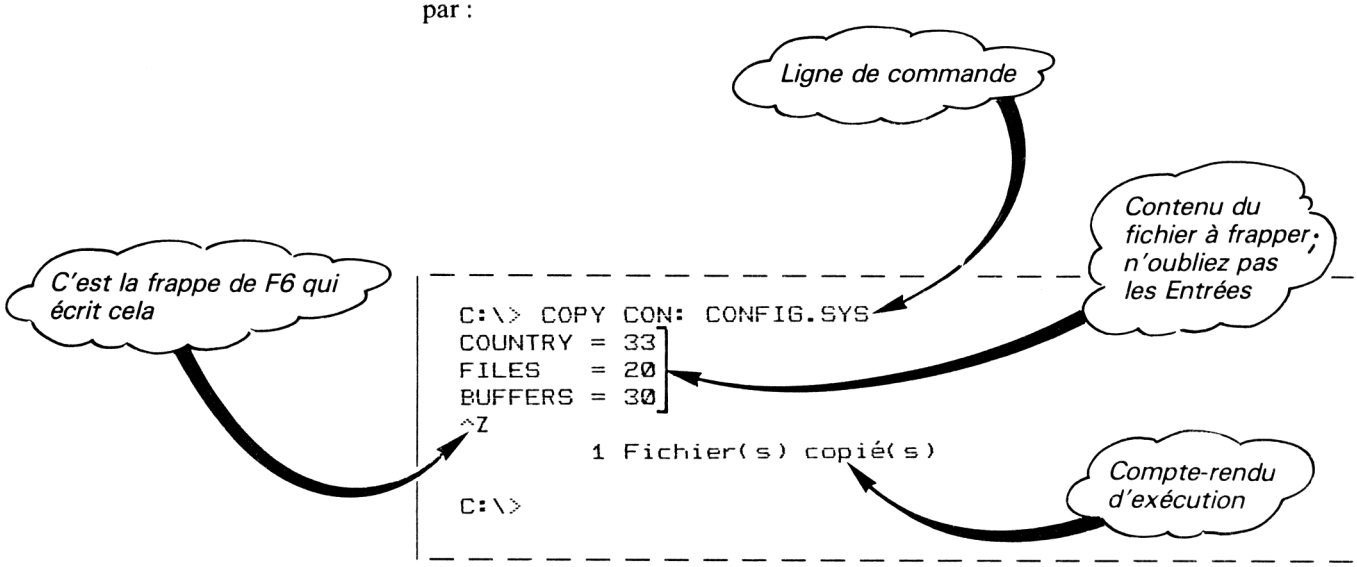
· La première consiste à recréer un nouveau fichier **CONFIG.SYS** , en refrappant le contenu de l'ancien, plus la nouvelle ligne. On a examiné comment procéder, pour cela, dans la fiche précédente ; aussi va-t-on le résumer brièvement ici.

C'est par cette méthode, également, qu'on crée un fichier neuf, par exemple CONFIG.SYS, s'il n'existait pas déjà.

Notez que les mêmes processus s'appliquent aux fichiers **AUTOEXEC.BAT** , ou autres ; il suffit de modifier les noms.

· La seconde méthode consiste à ajouter la seule ligne utile, *en recopiant* le contenu précédent.

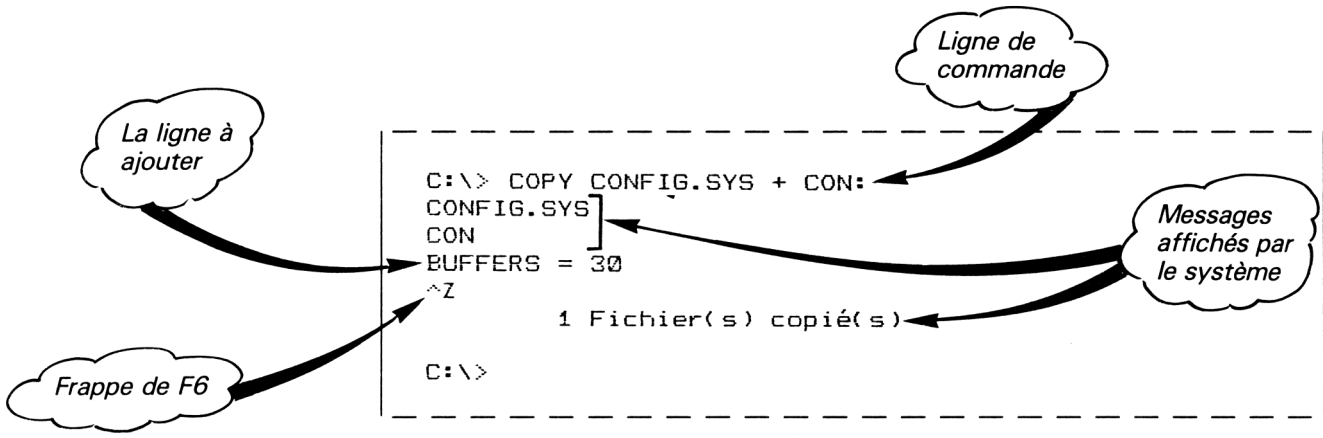
Commençons par la ré-écriture complète. On va ordonner à la machine de *copier* (**COPY**) ce que l'on frappe sur *la console* (**CON:**) dans le fichier **CONFIG.SYS** . Après l'*Entrée*, le système attend ce que vous aller taper au clavier ; on frappe les trois lignes (les deux pré-existantes, plus la nouvelle). N'oubliez pas les *Entrées* après chaque ligne. On termine la séquence avec la frappe de la touche F6, qui affiche **^Z** à l'écran (on aurait pu frapper directement sur *Ctrl*, maintenir la touche enfoncée, puis frapper sur Z, mais c'est plus long !). Le système *enregistre* ce fichier, fournit un compte-rendu d'exécution, puis se remet à vos ordres. Tout cela se traduit, à l'écran, par :



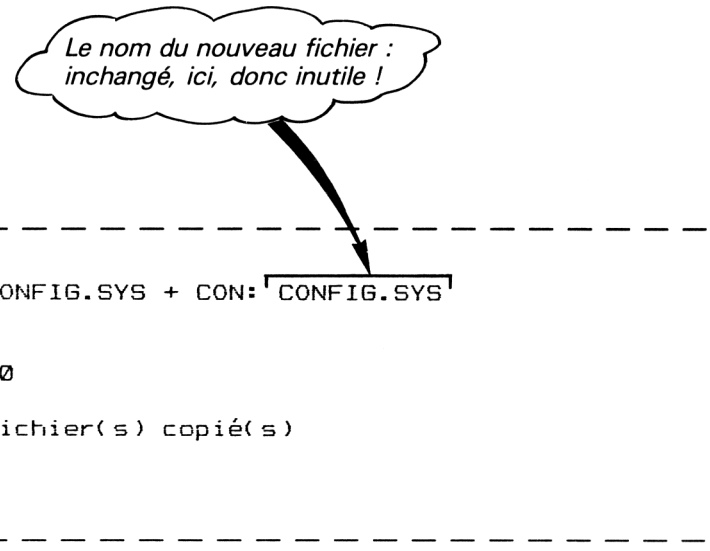
Un **TYPE CONFIG.SYS** vous prouvera que le fichier est bien complet :

```
C:\> TYPE CONFIG.SYS
COUNTRY = 33
FILES   = 20
BUFFERS = 30

C:\>
```

Voici comment ajouter une seule ligne. On est reparti du fichier *initial*, avec ses deux lignes. Cette fois, la commande spécifie qu'on veut *copier* (**COPY**) le fichier **CONFIG.SYS** mais qu'on va y ajouter (**+**) ce qu'on va frapper *au clavier* (**CON:**). Dans ce cas, et au lancement de la commande, le système recopie *l'ancien* CONFIG.SYS et l'écrit à l'écran, puis indique à l'écran qu'il attend vos frappes (**CON**). Il suffit alors de frapper la seule ligne **BUFFERS = 30** , puis de terminer sur F6 qui affiche **^Z** comme précédemment :



Le nouveau fichier est enregistré, vous pouvez le vérifier avec un **TYPE CONFIG.SYS** .
En variante de cette commande *avec concaténation* (le symbole « *plus* » !), certains opérateurs aiment spécifier, le nom *du nouveau* fichier. C'est inutile, ici, puisqu'il ne change pas, mais voici néanmoins ce que cela donne :



2. AVEC EDLIN

EDLIN

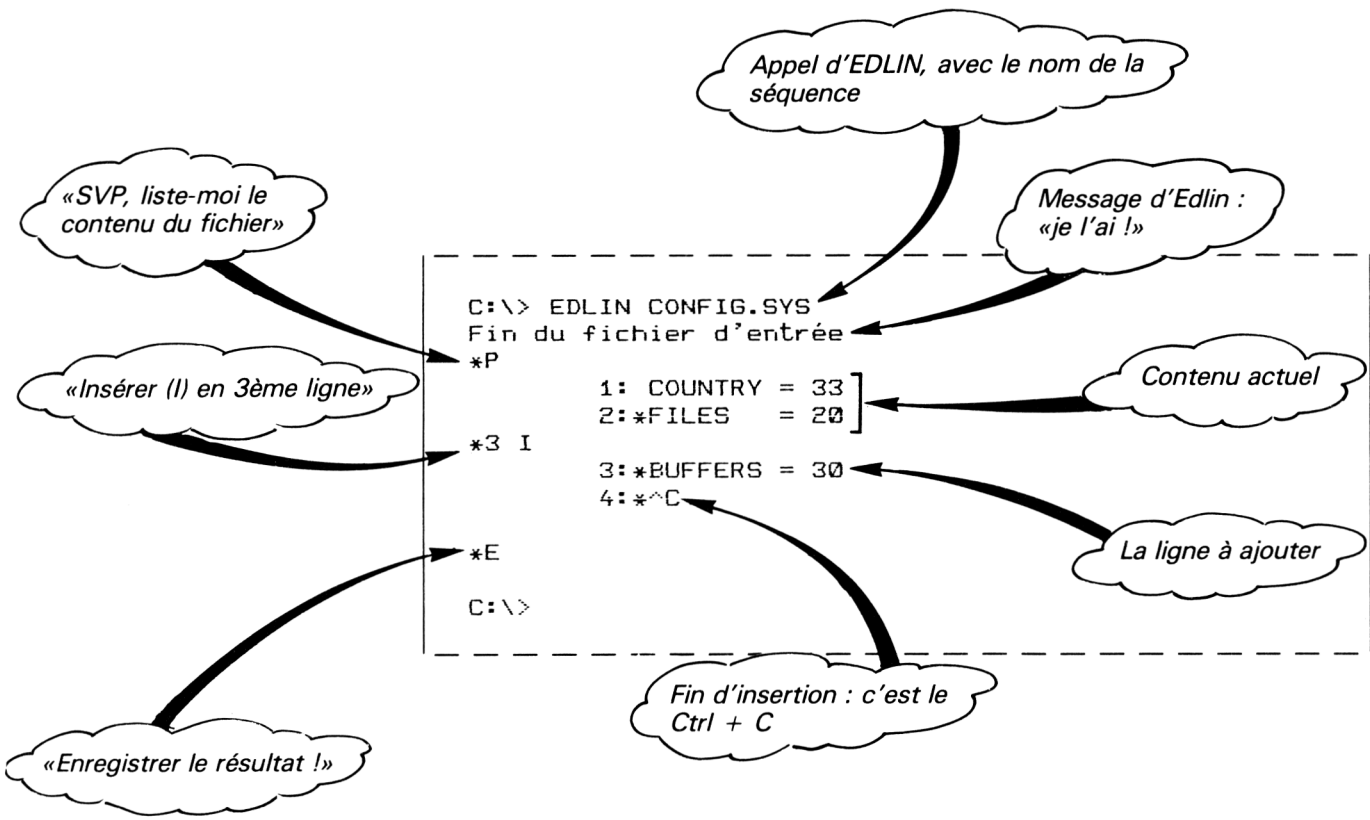
Programme *externe* du DOS, **EDLIN** est un « éditeur de lignes ». En fait, il s'agit d'un traitement de texte ultra-réduit, ne sachant travailler que ligne par ligne, et *un peu* plus difficile à utiliser qu'un bon traitement de texte tel qu'il en existe actuellement. Il permet, toutefois, de modifier totalement le contenu d'un fichier, si on le désire (suppressions, ajouts, déplacements, etc.).

Pour l'appeler, il faut qu' **EDLIN** soit présent dans votre répertoire actif, ou l'un des sous-répertoires alternatifs. On fait suivre son appel du nom de la séquence à traiter, ici **CONFIG.SYS** , reprit sous sa forme initiale. Notez qu' **EDLIN** a la manie de numéroter les lignes ; on se servira, ensuite, de ces numéros pour nos différentes manœuvres, mais ceux-ci disparaîtront dans le fichier définitif enregistré ; les astérisques (*) qui figurent ci-dessous sont dûs également à Edlin, n'en tenez donc pas compte.

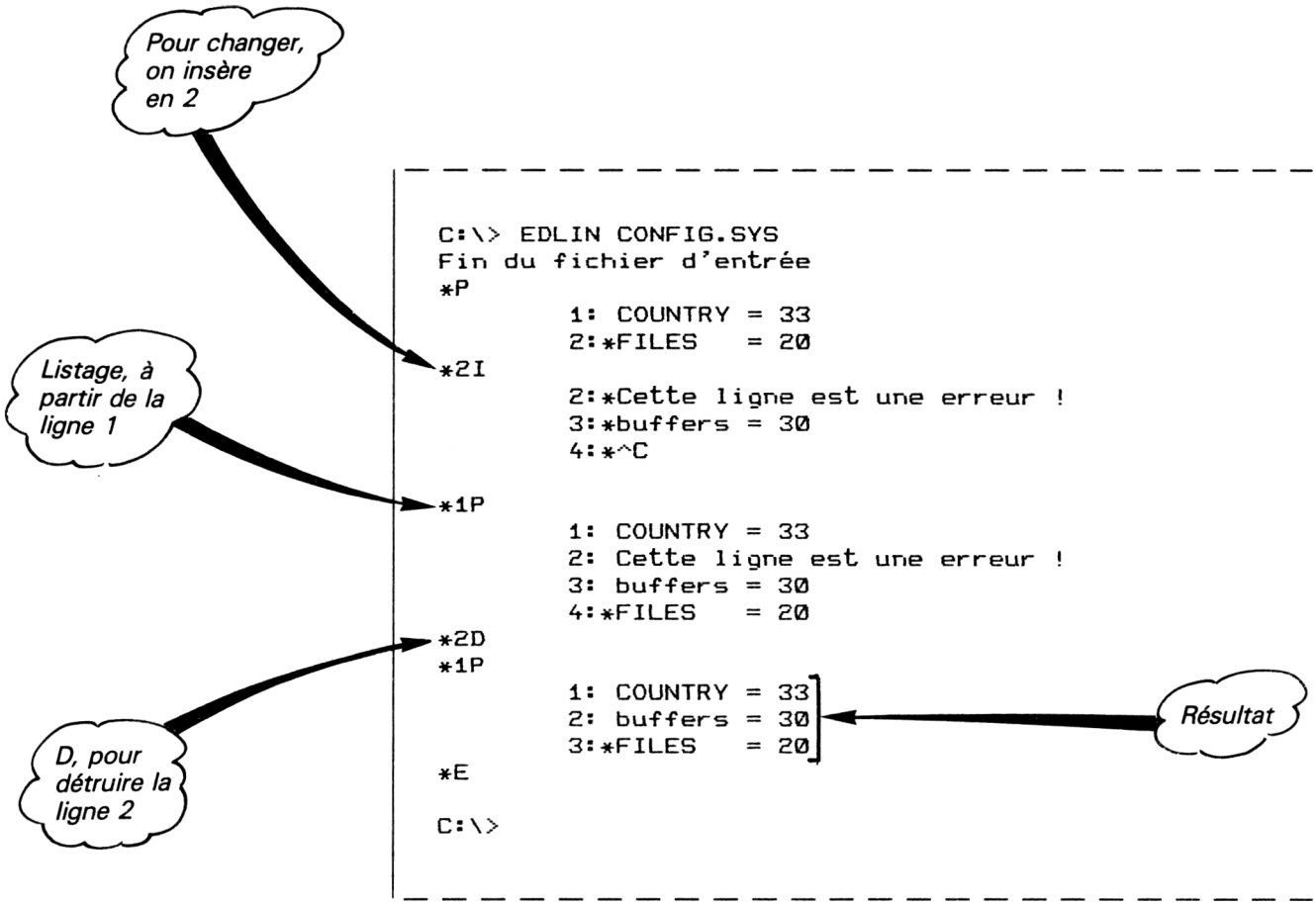
Les commandes à frapper vont être :

- **EDLIN CONFIG.SYS** pour l'appel.
- **P** , pour qu'il « publie », liste le contenu actuel du fichier.
- **3I** (ou **3i**) pour qu'il insère une nouvelle ligne, en *troisième* position.
- *Ctrl* + *C* (maintenez *Ctrl* enfoncée et frappez sur C), pour signifier la *fin de l'insertion.* Un **^C** apparaît à l'écran.
- **E** , pour enregistrer le résultat.

Voici donc la séquence complète :



Si vous vous trompez, quittez *Edlin sans enregistrer* le résultat, en frappant **Q** et non **E** , en fin de session. Ou alors, corrigez, en vous inspirant de l'exemple suivant où l'on a inséré, en 2, une ligne stupide :



Edlin permet encore bien des choses, mais vous en savez assez, ici tout au moins.

3. AVEC UN TRAITEMENT DE TEXTE

Vous pouvez employer n'importe quel traitement de texte pour modifier le fichier, *à la seule condition qu'il soit capable de fournir un texte en code ASCII pur.* Ce qu'ils sont pratiquement tous en mesure de faire. Vérifiez, sur le manuel de celui que vous utiliserez, comment procéder pour « sortir en ASCII », après avoir ajouté la ligne utile.

N'oubliez pas qu'il faudra ré-initialiser l'ordinateur afin qu'il lise le nouveau fichier **CONFIG.SYS** ou **AUTOEXEC.BAT** , si c'est celui-ci que vous avez modifié. Notez que ces mêmes séquences vous serviront à créer de *nouveaux* fichiers, si nécessaire.

35

Comment enregistrer dans un fichier spécifique un répertoire, ou toute autre information

Problème

On veut enregistrer « l'image » du répertoire, telle qu'elle apparaît à un instant donné, dans un fichier sur disque. Faut-il le refrapper dans ce nouveau fichier ?

Solution

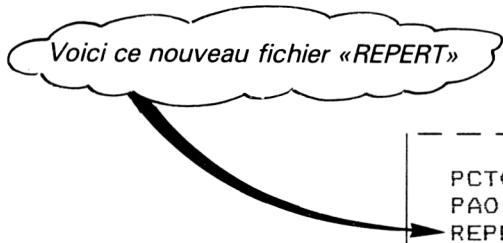
REDIRECTION

Il arrive parfois qu'on ait besoin d'enregistrer, sous forme d'un fichier, des informations qu'on veut figer à un instant donné. L'exemple type est le répertoire d'un disque : on veut, non seulement l'afficher et/ou l'imprimer, mais le stocker sous forme d'un fichier qu'on appellera, par exemple, REPERT. Cette expérience est intéressante à un autre titre : elle vous initiera, si ce n'est déjà fait, au concept de la *redirection.*

Le principe en est le suivant : on va commander un DIR mais, au lieu de le laisser bêtement s'afficher à l'écran, on va ordonner au système de *rediriger* vers un fichier ce qui aurait dû y apparaître (à l'écran). Le symbole de la *redirection* est > , la pointe de ce signe désignant le destinataire, tout simplement le nom du fichier choisi. Ainsi, et pour poursuivre notre exemple, on frappera :

```
C:\> DIR > REPERT
```

Notez que les espaces, autour du symbole >, sont facultatif. Le système enregistre aussitôt le répertoire dans le fichier REPERT, *ne l'affiche pas à l'écran*, puis ramène les caractères d'appel du DOS. Par contre, et dans le *nouveau* répertoire, on trouvera ce fichier ; voici un extrait de celui de l'auteur :




```
PCTOOLS      <DIR>      17/07/87    1:14
PAO          <DIR>      17/08/87    9:54
REPERT           1156   28/08/87   10:25
MP           <DIR>      22/07/87    8:54
XTREE        <DIR>       1/08/87    2:12
```

Ce que vous devez retenir ici, c'est l'usage du symbole de *redirection*, qui se révélera souvent fort pratique pour bien des manipulations.

36

Un court fichier à créer ou à modifier ? Utilisez EDLIN

Problème

Il est parfois nécessaire de changer une ligne dans un fichier, de le modifier autrement, ou encore d'en créer un nouveau. C'est le cas typique des deux fichiers incontournables du DOS, le `CONFIG.SYS` et `L'AUTOEXEC.BAT` ; *c'est également valable pour tous les fichiers* `.BAT` *que nous vous engageons à créer dans ce livre. Comment procéder à de telles modifications (ou créations) rapidement, simplement et efficacement ?*

Solution

EDLIN

Presque tous les progiciels de traitement de texte courants pourront servir à créer ou modifier de courts fichiers. Cela, *à la condition qu'ils sachent sortir en code ASCII* ; vérifiez-le pour le vôtre. Toutefois, il est souvent inutile de faire appel à un traitement de texte pour une création ou une modification bénigne. Il sera souvent plus simple de passer par un programme du DOS appelé `EDLIN` , pour *« éditeur de lignes »*, qui n'est autre qu'un petit traitement de texte rapide et efficace *pour de petits travaux.* Son défaut majeur, c'est qu'il ne sait travailler que sur une ligne à la fois, d'où son nom.

EDITEUR DE LIGNES

A quoi vous servira-t-il ? A modifier votre séquence `AUTOEXEC.BAT` , par exemple : ajouter une nouvelle ligne, en supprimer ou en modifier une, etc. Il interviendra également pour créer des séquences `.BAT` et pour tous les petits travaux de gestion courants, impliquant la rédaction de quelques mots ou lignes de texte en code ASCII pur. Nous vous déconseillons d'utiliser `EDLIN` pour des tâches plus importantes, par exemple la rédaction d'une lettre (encore que ce soit parfaitement faisable puisqu'Edlin peut traiter un document comportant jusqu'à 65 529 lignes !).

Ce programme comporte un certain nombre de commandes, dont quelques-uns sont des plus utiles ; elles sont déclenchées par la frappe de leur initiale, éventuellement accompagnée d'une indication de position dans le texte.

· `I` *sert à insérer* une ou plusieurs lignes. On indiquera, par exemple, `7I` pour insérer des lignes à partir de la septième.

Les commandes peuvent s'écrire en majuscules ou minuscules.

· *Ctrl* + *C* (touche *Contrôle* enfoncée, plus frappe de C) qui se traduit par l'affichage des caractères ^C , fait sortir du mode *« insertion »*.

COMMANDES POUR EDLIN

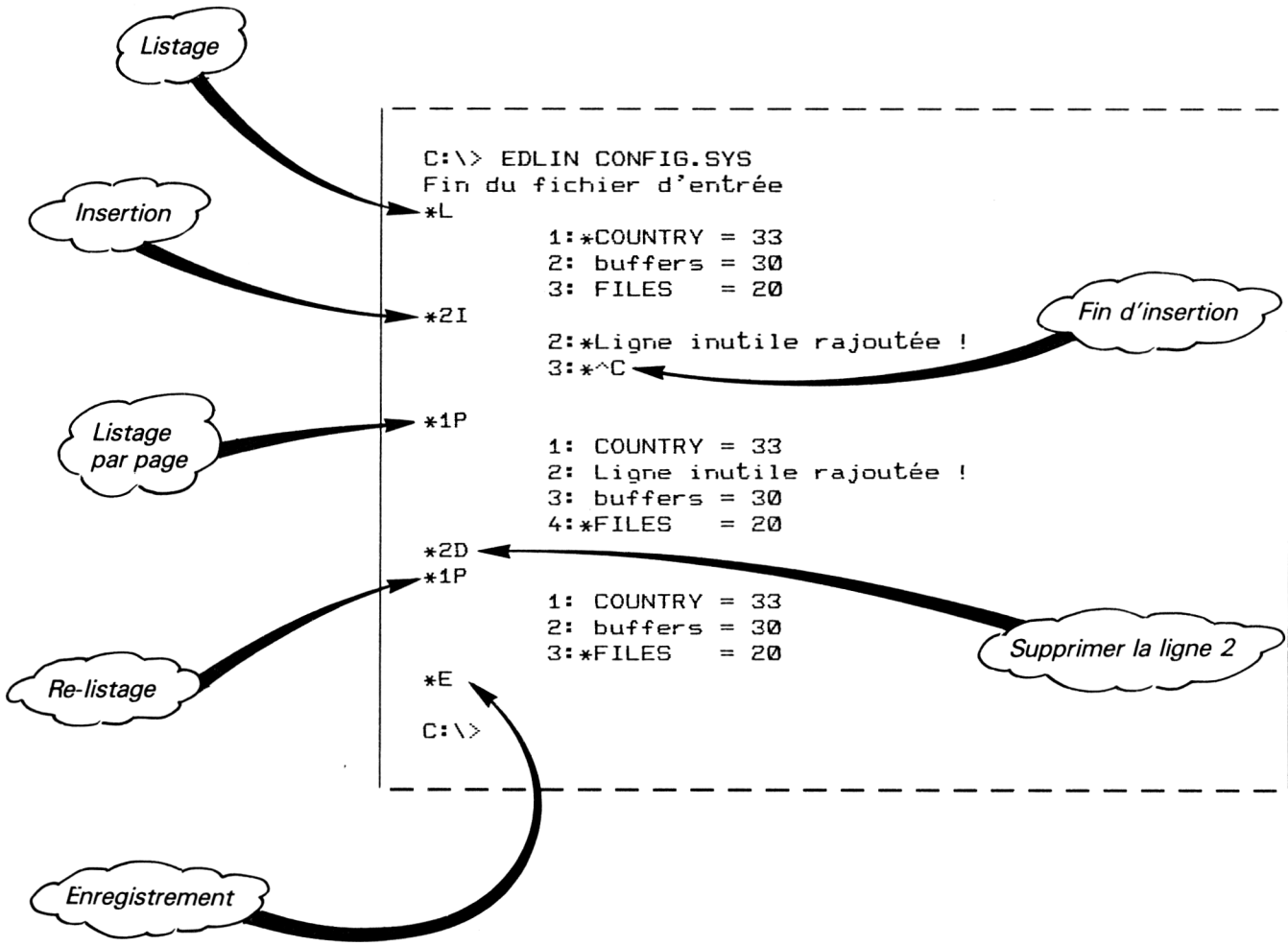
Commandes		Code	Fonction
Anglais	*Traduction*		
Append Lines	Ajout de lignes	A	Ajoute le nombre spécifié de lignes du disque au fichier en mémoire.
Copy Lines	Copie de lignes	C	Duplique des lignes.
Delete Lines	Suppression de lignes	D	Supprime des lignes.
Edit Line	Edition de ligne	Numéro ou point	Affichage de ligne.
End Edit	Fin d'édition	E	Sauvegarde d'un fichier.
Insert Lines	Insertion de lignes	I	Création, insertion de lignes.
List Lines	Listage de lignes	L	Listage de lignes.
Move Lines	Déplacement de lignes	M	Déplacement de lignes.
Page	Page	P	Listage de lignes
Quit Edit	Quitter l'édition	Q	Fin d'édition sans sauvegarde.
Replace Text	Remplacement de texte	R	Substitution de texte.
Search Text	Recherche de texte	S	Recherche de texte.
Transfer Lines	Transfert de ligne	T	Transfert d'un fichier dans un autre.
Write Lines	Ecriture lignes	W	Sauvegarde de lignes.

· **D** *sert à effacer « delete »* une ou plusieurs lignes. Par exemple, **7D** efface la 7e ligne ; ou encore **7,12 D** efface les lignes de 7 à 12.
· *Edlin numérote les lignes*, à l'écran seulement car ces numéros ne sont pas enregistrés dans le fichier. La frappe d'un numéro *seul* appelle la ligne visée, laquelle peut alors être modifiée.

· **L** *liste les lignes du fichiers.*
· **P** *en fait autant, par pages.* On indiquera **1P** pour lister à partir de la ligne **1** .
· **E** *enregistre* le nouveau fichier et fait revenir sous DOS.
· **Q** *fait quitter* Edlin *sans rien enregistrer* et ramène sous DOS.

Si vous voulez connaître toutes ses commandes, référez-vous au tableau d'accompagnement. Enfin, sachez que le caractère d'appel d'Edlin est un simple astérisque placé en début de ligne (de commande) ; les lignes de texte sont, elles, en retrait ; on y trouvera d'autres astérisques placés par Edlin afin de marquer, à chaque fois, la ligne active.

Pour invoquer Edlin, on frappe son nom, suivi du nom du fichier à créer, ou à appeler aux fins de modifications. Si le fichier n'existe pas, Edlin l'indique en écrivant *« Nouveau fichier »* ; s'il existe, il se place à la fin du fichier et écrit à l'écran *« Fin du fichier d'entrée »*. Dans l'exemple suivant, on a procédé à plusieurs manipulations types, en partant du CONFIG.SYS de l'auteur.



Essayez par vous-mêmes ces commandes minimales d'Edlin. Vous les maîtriserez très vite et apprécierez, ensuite, la rapidité avec laquelle on peut créer et modifier des fichiers.
N'hésitez pas, si vous le désirez, à entrer plus avant dans l'usage de ce programme en vous référant à un bon manuel du DOS.

37

Comment établir des chemins de recherche alternatifs pour les fichiers de données

Problème

Le DOS sait retrouver des programmes, dans quelque sous-répertoire qu'ils se trouvent si on a déclaré leur chemin au préalable. Mais pas les fichiers de données, apparemment. Que faire ?

Solution

La solution n'est applicable qu'à partir des DOS 3.2. Procurez-vous les si vous avez besoin d'exploiter cette possibilité.

Rappelons que les fichiers de programmes, *ceux se terminant par l'extention* `.COM` , `.EXE` ou `.BAT` , peuvent être retrouvés par tous les DOS, à partir des 2.Ø. Il suffit, pour cela, d'introduire dans le fichier `AUTOEXEC.BAT` une ligne `PATH` (*« chemin »*) spécifiant les chemins éventuels de rechange menant aux sous-répertoires que le DOS devra explorer, dans l'ordre, jusqu'à ce qu'il retrouve le programme voulu. Cela donnerait, par exemple, dans l' `AUTOEXEC.BAT` (exemple réel de l'auteur !) :

```
PATH=\;\DOS;\WINDOWS;\PM
```

Hélas, `PATH` ne fonctionne pas avec des fichiers dont l'extension n'est ni `.COM` , ni `.EXE` ni `.BAT` . Or, il est parfois bien pratique de ranger certains fichiers utilisateurs dans un sous-répertoire spécifique, ne serait-ce que par souci de bonne organisation. Si l'on recherche alors un fichier s'appelant, par exemple, `CLIENTS.SUD` , se trouvant dans le sous-répertoire `CLIENTS.FRA` , le DOS *n'ira pas* explorer les sous-répertoire, quand bien même on aurait déclaré leurs chemins avec `PATH` .

APPEND

Les DOS à partir des 3.2 pallient cette lacune. En effet, ils introduisent une commande symétrique de **PATH** , agissant pour les fichiers de données, cette fois, et appelée **APPEND** . On peut vérifier son existence avec un **DIR** :

```
C:\DOS> DIR APPEND

 Volume dans unité C est LILEN
 Répertoire de  C:\DOS

APPEND   EXE      5873  18/03/87  12:00
        1 Fichier(s)
  1155072 octets disponibles

C:\DOS>
```

Cette commande s'emploie tout comme **PATH** , avec la même syntaxe. On écrira dans l' **AUTOEXEC.BAT** et juste après le **PATH** , par exemple :

```
C:\> TYPE AUTOEXEC.BAT
PATH \;\DOS;\WINDOWS;\PM
APPEND \CLIENTS.FRA;\COURRIER
KEYBFR
DATE
TIME
graphics
PROMPT $P$G
DIR *.
REM Bonjour ! Faites votre choix.

C:\>
```

Voici le APPEND

Dès lors, vous aurez également accès à ces sous-répertoires pour les fichiers de données. Voyez notre fiche *« Ajouter des lignes dans* **AUTOEXEC.BAT** *ou* **CONFIG.SYS** *? Voici comment »* si vous ne savez comment introduire la nouvelle ligne dans l' **AUTOEXEC.BAT** .

A l'initialisation du micro, et toujours avec les mêmes exemples, l'écran affichera :

La preuve que APPEND est bien actif !

```
C>PATH \;\DOS;\WINDOWS;\PM

C>APPEND \CLIENTS.FRA;\COURRIER

C>KEYBFR

C>DATE
Date du jour : Mar  1/01/1980
Entrez la nouvelle date
(jj/mm/aa) :
```

Ou encore, et en cours de route, on vérifiera, sous DOS, quels sont les chemins actifs, avec **PATH** d'une part, et **APPEND** d'autre part :

```
C:\> PATH
PATH=\;\DOS;\WINDOWS;\PM

C:\> APPEND
APPEND=\CLIENTS.FRA;\COURRIER

C:\>
```

Répétons-le : **APPEND** n'existe qu'à partir des DOS 3.2.

38

Comment introduire des paramètres à la demande en lançant des fichiers .BAT

Problème

On a créé des fichiers `.BAT` *pour automatiser certaines séquences. Or, on souhaite être « dirigiste » et pouvoir fournir au fichier* `.BAT` *des informations à leur lancement qui spécifieront leur action. Prenons un exemple simple : on lance l'affichage d'un volumineux répertoire ; on chaîne la commande avec un* `FIND` *, pour sélectionner certains fichiers et pas d'autres. Selon les besoins, ce seront les fichiers* `.COM` *, ou* `.EXE` *, ou* `.FON` *, etc. Comment procéder ?*

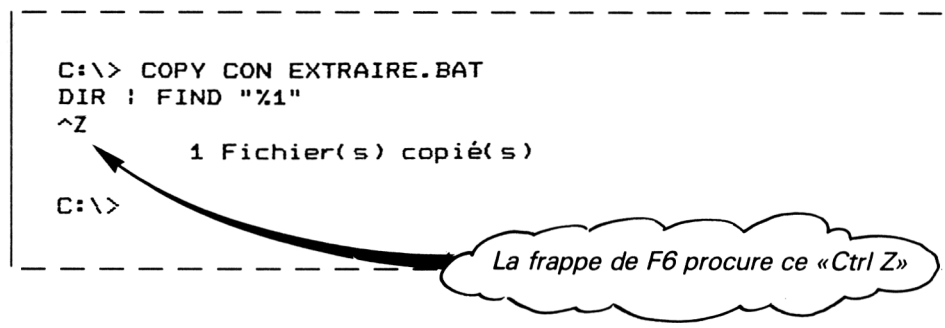
Solution

Ce très simple exemple va vous permettre de bien comprendre la démarche ; bien sûr, ensuite, vous pouvez l'appliquer à des besoins plus complexes.
D'abord, on va créer un fichier `.BAT` appelé `EXTRAIRE.BAT` . Il ne contient qu'une ligne, avec un `DIR` chaîné avec le `FIND` (nous avons consacré une fiche spécifique à l'usage de `FIND`).

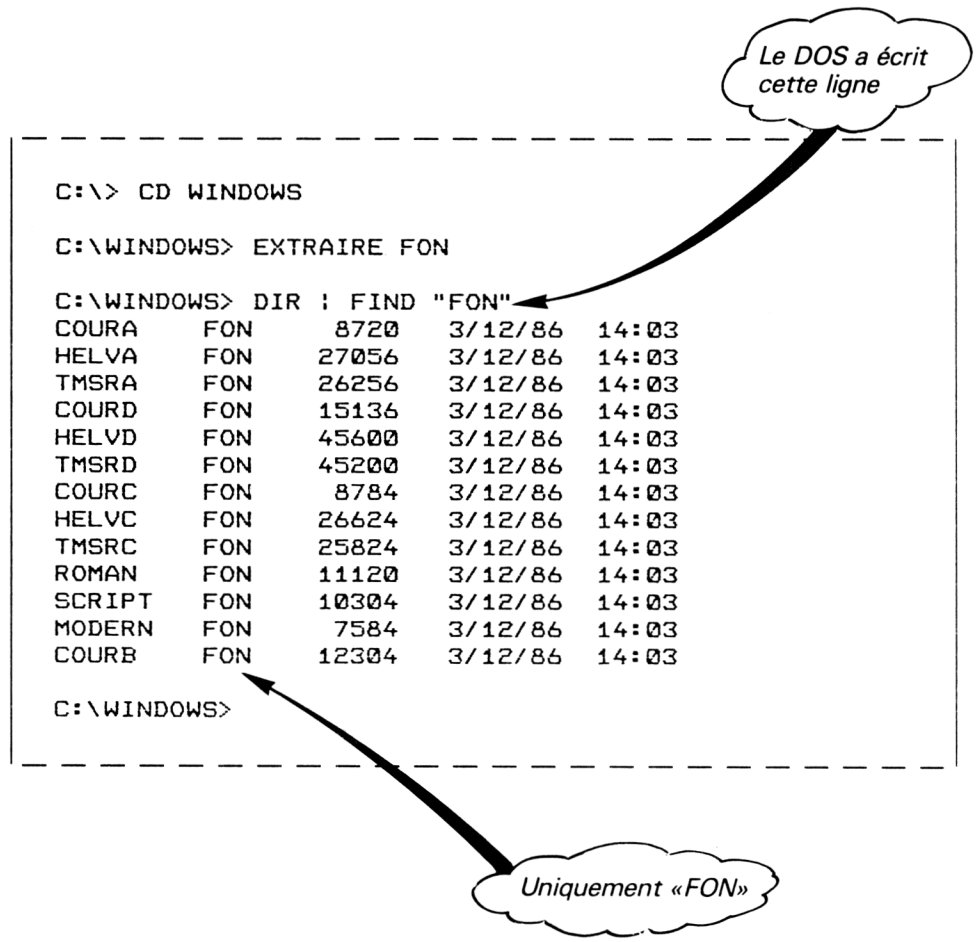
Rappelez-vous que pour chaîner des commandes, il faut les séparer d'une barre verticale coupée. C'est le code ASCII numéro 124, obtenu en maintenant *Alt* enfoncée et en frappant 124 sur le *seul* clavier numérique (de droite).

PARAMETRE REMPLAÇABLE

Toute l'astuce va consister à faire suivre le **FIND** d'un *paramètre remplaçable*, le paramètre numéro 1 (car on peut en aligner une dizaine, et même davantage). Dans le langage « dossier », *paramètre* s'écrit *pourcent* (%) ; on va introduire **"%1"** obligatoirement entre guillemets, ce qui mène au fichier **EXTRAIRE.BAT** suivant (une fiche explique comment construire aussi un fichier **.BAT** avec un **COPY CON** ; frappez simplement ce qui suit, ici) :



Afin de vous montrer ce que procure ce fichier, l'auteur est passé à son sous-répertoire WINDOWS, fort bien garni. Pour connaître tous les fichiers de fontes de caractères (pour le traitement de texte ou la PAO), il suffit d' **EXTRAIRE.FON** , où **FON** est l'extension des noms de fichiers des polices de caractères. Pour le système, **FON** est le paramètre qui va remplacer le **"%1"** ; il l'écrit d'ailleurs aussitôt, puis liste les seuls fichiers contenant **FON** :



Veut-on les fichiers **.COM** ? Voici :

```
C:\WINDOWS> EXTRAIRE COM

C:\WINDOWS> DIR ¦ FIND "COM"
WIN      COM     4969   3/12/86  14:01

C:\WINDOWS>
```

Ou les **.EXE** ? Voyez :

```
C:\WINDOWS> EXTRAIRE EXE

C:\WINDOWS> DIR ¦ FIND "EXE"
SPOOLER  EXE    13392   3/12/86  14:02
MSDOS    EXE        1  23/04/87  14:50
AGENDA   EXE    37984   3/12/86  14:00
BLOCNOTE EXE    19040   3/12/86  14:00
CALC     EXE    25040   3/12/86  14:00
CONTROL  EXE    53424   3/12/86  14:00
HORLOGE  EXE     7984   3/12/86  14:00
PAINT    EXE    90512   3/12/86  14:00
PRESSPAP EXE     9712   3/12/86  14:00
REPERT   EXE    37376   3/12/86  14:00
REVERSI  EXE    14864   3/12/86  14:00
TERMINAL EXE    47664   3/12/86  14:00
WRITE    EXE   192880   3/12/86  14:03
EXERCICE WRI     3072   3/12/86  14:03

C:\WINDOWS>
```

Remarquez la dernière ligne : EXE est contenu dans EXERCICE ! Le DOS n'a pas fait la distinction, avec **FIND** . Méfiez-vous de ce type de piège.

Maintenant, transposez cela à des fichiers de clients, de pièces, etc. Si vous multipliez le nombre de paramètres remplaçables, y compris avec des commandes différentes, utilisez les notations **%1** , puis **%2** , puis **%3** , etc. Si le symbole **%** entre lui-même dans le nom, doublez-le (**%%**).

39

Comment redater un fichier

Problème

On dispose d'un fichier portant une date et une heure incorrectes. On veut le redater.

Solution

REDATER UN FICHIER

COPY

Redater un fichier impose qu'on passe par COPY, en faisant, toutefois, une fausse copie s'il doit rester sur le même disque. Supposons qu'on dispose d'un fichier CLIENTS dans le sous-répertoire TRANSFER :

```
C:\TRANSFER> dir

 Volume dans unité C est LILEN
 Répertoire de  C:\TRANSFER

.            <DIR>      14/07/87  12:32
..           <DIR>      14/07/87  12:32
CLIENTS            32   1/08/87   9:30
        3 Fichier(s)
  1040384 octets disponibles

C:\TRANSFER>
```

La date du fichier CLIENTS, le 1-8-87, ainsi que l'heure, ne conviennent pas. On est désormais (date du système : mais on peut la modifier à volonté) le 23-10-87. Si l'on copie ce fichier sur **A:** , par exemple, vérifiez que sa date *n'est pas* modifiée :

```
C:\TRANSFER> COPY CLIENTS A:
        1 Fichier(s) copié(s)

C:\TRANSFER> DIR A:

 Volume dans unité A est GESTION
 Répertoire de  A:\

CLIENTS            32   1/08/87   9:30
        1 Fichier(s)
   361472 octets disponibles

C:\TRANSFER>
```

Bien qu'on ait fait la copie au 23-10-29 à 18 h, rien n'est changé

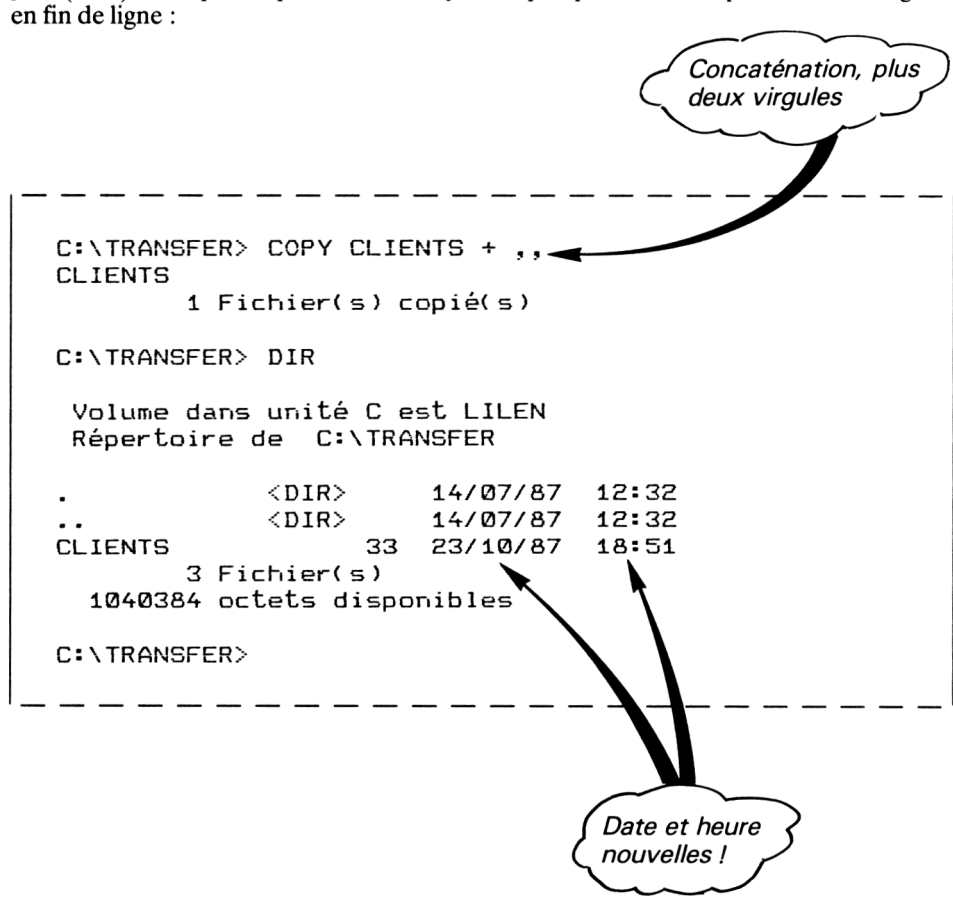
Pour recopier en *redatant*, il faut ajouter le symbole de la *concaténation*, un *plus* (**+**) à la commande :

Notez ce +, en plus !

```
C:\TRANSFER> COPY CLIENTS A: +
CLIENTS
        1 Fichier(s) copié(s)

C:\TRANSFER> DIR A:

 Volume dans unité A est GESTION
 Répertoire de  A:\

CLIENTS            33  23/10/87  18:49
        1 Fichier(s)
   361472 octets disponibles

C:\TRANSFER>
```

Cette fois, les date et heure ont changé

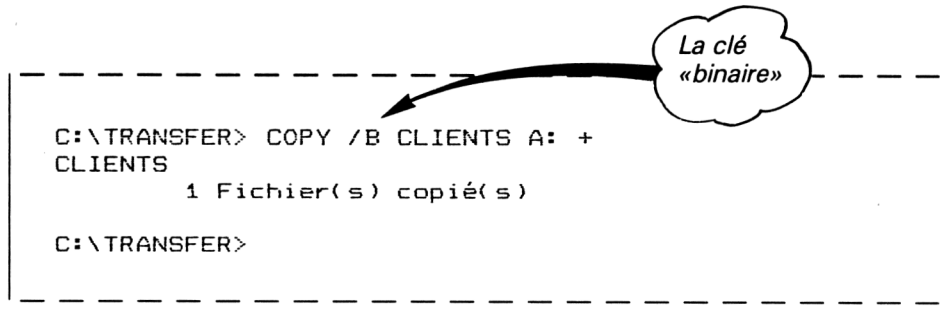
Pour redater un fichier *sans le déplacer*, on fait semblant de la concaténer, avec un *plus* (**+**), mais parce qu'il faut alors ajouter quelque chose, on placera *deux* virgules en fin de ligne :



Ci-dessus, il n'y a pas réellement recopie, mais redatage.

Attention toutefois à un piège horrible : tout ceci ne marche qu'avec des fichiers en *ASCII pur*. La copie (avec **COPY**) s'effectue jusqu'au caractère de *fin de fichier*, qui est ce *Contrôle Z* (obtenu via a frappe de F6, par exemple). En ASCII, *Contrôle Z* est le caractère de code 26, décimal. Or, si le fichier est en binaire, ou contient des codes binaires (cas des fichiers issus de traitements de texte, de tableurs, etc.), il pourra s'y trouver ce fameux code 26, avec une toute autre signification que celle de fin de fichier, ce que **COPY** ignore : la copie cessera et tout ce qui suit sera perdu !

A moins qu'on ne prévienne **COPY** , en lui disant qu'il s'agit d'un fichier *binaire* à l'aide de la clé **/B** . Dans ce cas, la copie ne s'arrête plus au caractère *Ctrl Z* mais porte sur tout le fichier, selon sa longueur enregistrée dans le répertoire. La commande est donc, pour une copie vers **A:** , par exemple :



N'oubliez donc pas ce point important, qui explique qu'on ait réservé cette fiche aux *experts*.

40

Comment imprimer ce qui apparaît à l'écran ?

Problème

On veut imprimer ce qui apparaît à l'écran, sélectivement ou non. Quelles sont les méthodes applicables ?

Solution

RECOPIE D'UN ECRAN

SHIFT + PRTSC

Pour imprimer ce qui apparaît à l'écran, il existe plusieurs méthodes (mais n'oubliez pas de mettre l'imprimante en marche avant de les expérimenter !) :

1) — La recopie d'un écran, tel, est obtenue en frappant sur *Shift* + *PrtSc* (maintenez la touche de passage instantané aux majuscules, *Shift*, enfoncée, et pressez la touche *PrtSc* (*« Print Screen », impression écran*), notée encore *Impr Ecr* sur les claviers francisés.

Attention :
· Avec certains claviers 102 touches, la seule frappe de « Impr Ecran » (ou PrtSc) suffit.
· Certains programmes d'applications reprogramment les touches à leur façon. Donc, parfois, *Shift* + *PrtSc* ne commanderont plus une recopie d'écran.
· Si l'écran contient des caractères « graphiques », autres que les habituels caractères alphabétiques, il faudra avoir chargé au préalable le fichier `GRAPHICS` afin de les reproduire.

Les plus fréquents de ces caractères graphiques sont ceux servant à tracer des cadres, des filets ou des traits. Sans **GRAPHICS** , ils ne seront pas reproduits par l'imprimante. Pour charger ce programme « externe », appartenant au DOS, on frappe simplement, sous DOS :

```
C:\> GRAPHICS

C:\>
```

Si vous deviez reproduire des écrans avec des caractères graphiques, introduisez tout simplement **GRAPHICS** dans votre **AUTOEXEC.BAT** .

2) — Lorsqu'on veut afficher non plus simplement un écran, mais tout ce qui va y défiler, on met l'imprimante en service en permanence avec un *Ctrl* + *PrtSc* : maintenez la touche *Contrôle* (*Ctrl*) enfoncée et frappez sur *PrtSc* (*« Print Screen », impression écran*), notée *Impr Ecr* sur les claviers francisés.

CTRL + PRTSC

Dès lors, tout le dialogue apparaissant à l'écran va être imprimé, *jusqu'à une nouvelle frappe de la même combinaison, Ctrl* + *PrtSc*, qui annule l'ordre précédent.

CTRL + P

On peut généralement frapper *Crtrl* + *P* (maintenez *Contrôle* enfoncée et frappez la lettre P) à la place de *Ctrl* + *PrtSc*, avec le même effet.

Ce mode d'impression, assimilable à un mode « perroquet », car tout ce qui se passe à l'écran est reproduit, est dit *impression en écho.*

3) — Une troisième méthode, plus sélective mais aussi plus savante, consiste à « rediriger » vers l'imprimante ce qui est listé à l'écran. En voici un exemple :

· *Si l'on commande* **DIR** , l'écran va afficher le contenu du répertoire actif.
· *Si l'on veut imprimer ce contenu,* on va ordonner au système de *rediriger* ce qui apparaît à l'écran *vers* l'imprimante. Le symbole de *redirection* est **>** , la pointe étant orientée vers le destinataire.

PRN

LPT1

L'imprimante s'appelle, pour le système :
· PRN , pour « *Printer* », imprimante, de façon très générale ; ou encore :
· LPT1 , car elle est souvent connectée au port parallèle de sortie numéro un.
Si votre DOS refuse PRN et LPT1 , ajoutez un double-point (:) à leur suite et sans espace : PRN: ou LPT1: .

Donc, pour imprimer le **DIR** , on le redirige sur l'imprimante (*il n'apparaîtra pas à l'écran !*), avec :

```
C:\> DIR > PRN
```

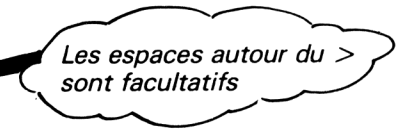
TYPE

REDIRECTION

De la même façon, on pourra imprimer le contenu complet d'un fichier, avec un **TYPE** (qui devrait l'afficher), suivi d'une *redirection* vers l'imprimante. L'auteur dispose d'un petit fichier appelé SALUTS ; son affichage à l'écran procure :

```
C:\> TYPE SALUTS
Bonjour, Madame.

C:\>
```

Avec la *redirection* suivante, le texte est *uniquement* imprimé, l'écran se bornant à afficher :




```
C:\> TYPE SALUTS > PRN

C:\>
```

Par contre, l'imprimante sort :

```
Bonjour, Madame.
```

Nous vous avons dit qu'on peut remplacer **PRN** par **LPT1** , avec le même résultat ; cela donnerait :

```
C:\> TYPE SALUTS > LPT1
```

Vous apprécierez cette troisième méthode dans bien des cas, dont le suivant : la disquette de certains progiciels comporte parfois un fichier appelé **LISEZ.MOI** , de plusieurs pages d'écran. Imprimez-le alors aussitôt en ordonnant :

```
C:\> TYPE LISEZ.MOI >PRN
```

Vous disposerez ainsi d'une copie papier, souvent, plus agréable à consulter.

41

Comment passer à l'impression en caractères condensés

IMPRESSION EN CARACTERES CONDENSES

Problème

On veut mettre l'imprimante en mode d'impression condensée, et ce, à volonté, à partir du DOS et sans toucher à la programmation interne de l'imprimante.

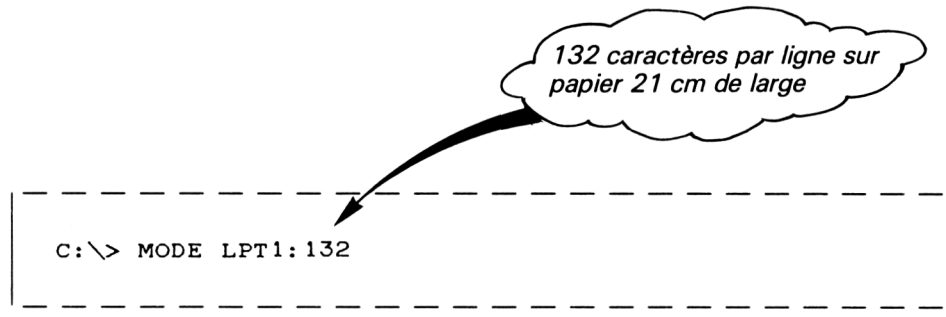
Solution

MODE

LPT1

La commande **MODE** , du DOS, permet une programmation très réduite de l'imprimante puisqu'elle n'affecte que le mode : *condensé* ou *normal.* Avec du papier standard de 21 cm de largeur, on peut imprimer jusqu'à 80 caractères par ligne en mode normal, 132 (à 136) en condensé ; avec du papier grande largeur de 36 cm, la ligne normale est de 136 caractères ; en condensé, elle passe à 232 caractères, et peut même atteindre 272 caractères selon la police utilisée. Cela permet l'impression d'états ou de tableaux de grande envergure.

Pour passer en mode condensé, sous DOS, on ordonne simplement (**LPT1** représente l'imprimante parallèle, connectée, au port parallèle 1) :



Le système enregistre l'ordre, avec l'*Entrée*, et l'imprimante passe en mode *condensé* ; la recopie d'écran procure en mode étroit :

```
C:\) MODE LPT1:132

LPT1: réglée à 132

C:\)
```

Le retour au mode normal s'effectue de façon symétrique avec :

```
C:\) MODE LPT1:80

LPT1: réglée à 80

C:\)
```

Hormis cette possibilité, rappelez-vous que, d'une façon générale, les imprimantes peuvent être programmées soit via des interrupteurs internes, soit grâce à un tableau de bord. En outre, elles disposent de codes de programmation qu'on peut leur adresser de diverses façons, en particulier à partir de bien des progiciels.

42

Continuez à travailler sur votre micro tandis que l'imprimante sort votre listage

Problème

En règle générale et lorsqu'on lance une impression, l'ordinateur reste immobilisé et ne peut rien faire d'autre. Ce qui est fort gênant si l'on est en train d'imprimer un listage de 120 pages sur une imprimante matricielle à 120 caractères par seconde. Or, on peut fort bien imprimer « en arrière-plan » et continuer à travailler sur la machine.

Solution

PRINT

MULTITACHE

Pour résoudre ce problème et pouvoir continuer à travailler sur le micro-ordinateur pendant que l'imprimante déroule ses pages, on va faire appel à un ordre du DOS qui est tout simplement **PRINT** (« imprimer »). C'est même le seul qui permette ce type de travail, dit *multitâche*, car une tâche s'exécute en *arrière-plan* (l'impression), tandis qu'en *avant-plan*, on continue à travailler en traitement de texte, par exemple.

Cette méthode s'applique essentiellement à des listages, ou à du courrier courant. Pour cela, il faut :

- *Sortir sur disque* ce qui va devoir être imprimé, en code ASCII.
- Revenir sous DOS.
- Lancer l'impression sous DOS.
- Puis repartir dans un programme quelconque.

L'impression lancée se poursuivra, sauf contre-ordre explicite. Bien sûr, et pendant que l'imprimante imprime les fichiers indiqués, vous ne pourrez exécuter des impressions inopinées.

Supposons qu'on veuille imprimer RELANCES.MAI, fichier rangé sur disque en ASCII. De retour sous DOS, on frappe **PRINT RELANCES.MAI** ; avec le premier PRINT (et uniquement avec lui), le DOS va vous demander si l'unité de sortie est bien

l'imprimante dénommée **PRN** (pour « *Printer* », imprimante). C'est l'imprimante connectée au port parallèle. On acquiesce en frappant simplement *Entrée* ; le système installe la partie résidente de PRINT et lance l'impression :

```
C:\> PRINT RELANCES.MAI

Nom de périphérique de liste [PRN]:
Partie résidente de PRINT installée

C:\>
```

Vous pouvez, maintenant, reprendre une tâche quelconque. Mais attention ! L'impression ira jusqu'à son terme : vous ne pourriez la stopper définitivement en cours de route qu'en frappant à nouveau un **PRINT** , sous DOS, suivi du code magique d'arrêt qui est :

```
C:\> PRINT /T
```

Avec l'*Entrée*, l'impression stoppe et l'écran affiche :

```
File d'attente d'impression vide
```

En cours d'impression, on peut ajouter d'autres fichiers à imprimer à la suite avec :

```
C:\> PRINT RELANCES.JUN /P
```

Le fichier RELANCES.JUN se placera dans la file d'attente d'impression. Pour retirer un fichier de cette file d'attente, on commandera :

```
C:\> PRINT RELANCES.AVR /C
```

En fait, on peut faire suivre **/P** et **/C** d'autres noms de fichiers, *séparés par un espace*. On peut mêler des **/P** et **/C** sur une même ligne, une clé restant active tant qu'une autre ne survient pas.

On peut user aussi des caractères de remplacement **?** et ***** et, par exemple, imprimer tous les fichiers RELANCES avec :

```
C:\> PRINT RELANCES.*
```

SPOOLER

Notez qu'une autre méthode, qui ne fait plus appel à **PRINT** , est possible. Elle consiste à créer une zone tampon en mémoire, un *buffer* ou un *spooler*, qui stockera le texte à imprimer et le dirigera vers l'imprimante au fur et à mesure des besoins. C'est ce qui se pratique d'office en *publication assistée par ordinateur* (PAO), lorsqu'on pilote une imprimante laser et qu'on ne veut pas bloquer par trop longtemps l'ordinateur. Dans ce cas, on peut imprimer des textes dans tous les formats, des graphiques, etc.

43

Comment programmer l'imprimante sous Debug

Problème

On veut programmer l'imprimante afin qu'elle imprime en gras, par exemple (ou en condensé, ou en...). Ce qui permettra d'obtenir de meilleures sorties, avec des programmes qui n'ont aucun pouvoir d'action sur ce périphérique.

Solution

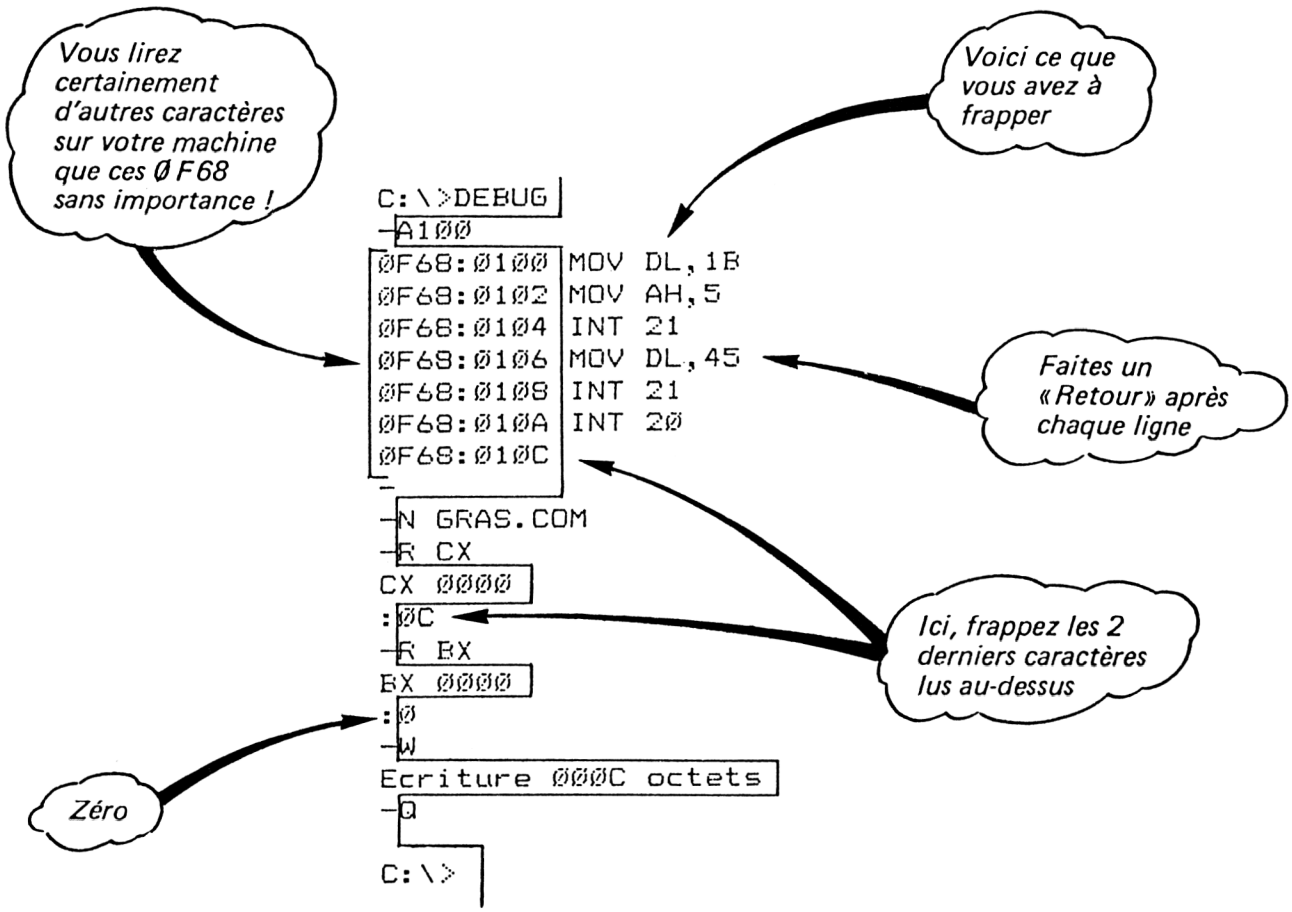
La plupart des traitements de texte offrent de vastes possibilités de programmation des imprimantes. Mais tel n'est pas le cas des autres progiciels, encore moins du DOS (qui permet toutefois de passer aux *caractères condensés*, voyez la fiche spéciale sur ce sujet). Eventuellement, vous pourrez programmer votre imprimante à l'aide de ses commutateurs internes, si c'est une matricielle (c'est le type d'imprimante le plus répandu), mais c'est fort désagréable si l'on doit modifier fréquemment la programmation et source de pannes. Certaines matricielles permettent une programmation par tableau de bord, mais malheureusement pour un nombre restreint de fonction. Aussi vous proposons-nous deux façons de procéder :

· On va rédiger un programme en langage d'assemblage, sous « Debug » (même si vous ne comprenez rien à l'assembleur, vous pourrez en venir à bout).
· Ou encore, on va en écrire un en Basic, mais ce sera l'objet de la fiche suivante.

Supposons qu'on veuille programmer l'imprimante pour qu'elle écrive **en gras**, qu'il s'agit d'une imprimante matricielle, et qu'elle respecte les codes *Epson*, utilisés par l'imprimante graphique *IBM*, par exemple. Dans ce cas, le code de mise en gras est *« ESC E »*, ce qui ne s'invente pas et est indiquée en toutes lettres dans son manuel.

Il nous faut « traduire » cet *Esc E* en numérique, et même en *hexadécimal* (cela, c'est pour le premier rang de la classe). On retrouvera ces codes hexadécimaux dans le même manuel de l'imprimante, parfois en cherchant bien. On lira alors que *Esc* se code 1B et que la lettre E, en code ASCII, s'écrit 45. Nous sommes prêt.

Vérifiez que le programme externe du DOS, appelé DEBUG, est bien disponible ; on va l'appeler. Son caractère d'appel est un simple tiret. Voici ce qu'il va vous falloir exécuter, tel que ça apparaîtra à l'écran. Nous avons délimité par un trait torturé ce que la machine affiche (à gauche), et ce que vous devez frapper (à droite). Les nombres **0F68**, à gauche, pourront être différents sur votre machine, sans aucune importance ; nous vous commentons cette séquence aussitôt après :



Avec la première ligne, on appelle Debug comme n'importe quel autre programme. Puis, on lui dit qu'on veut créer et « assembler » une séquence commençant en **100** avec un **A100**. Les 7 lignes qui suivent constituent cette séquence : frappez-les très exactement telles qu'elles apparaissent ici, sans oublier les virgules. Vous retrouverez, au passage, les codes **1B** puis **45**. Après quoi, on va *nommer* (avec **N**) ce programme, par exemple GRAS. Il faut y ajouter l'extension COM, donc écrire **GRAS.COM**. Suivent deux manipulations pour mettre le système en condition : frappez **0C**, répétition des deux derniers caractères lus au-dessus puis obligatoirement un zéro après R BX. La commande **W** (de *« Write », écriture*), déclenche l'enregistrement de ce document sur le disque, puis **Q** fait « quitter » Debug, qui a bien rempli son office.

De retour sous DOS avec **C:\>**, vous pouvez appeler cette séquence GRAS comme tout autre programme afin de la lancer :

```
C:\> GRAS
```

Désormais, votre imprimante est programmée en gras et écrira en gras. Vous pouvez inclure cette séquence dans un fichier BAT. Vous pouvez aussi préparer de la même façon des fichiers pour l'italique, la qualité courier, *et tout ce que votre imprimante permet* en modifiant :

· Les codes **1B** et **45** du gras, à remplacer par les codes que vous sélectionnez.
· Les premières 3 lignes (**MOV DL, XX** avec votre code, **MOV AH,5** puis **INT21**) sont obligatoires dans tous les cas. S'il n'y a qu'un seul code à émettre, terminez par un **INT20**).
· Chaque code supplémentaire exige deux lignes : un **MOV DL,XX** avec votre code, puis un **INT21** .

Voilà : c'est plus rapide à faire qu'à expliquer. Si vous vous perdez, faites **Q** et recommencez tout.

44

Comment programmer l'imprimante en Basic

Problème

On veut programmer l'imprimante afin d'écrire en gras, en condensé, etc., et ce, en passant par le Basic, et non par Debug comme dans la fiche précédente.

Solution

Le langage Basic permet tout aussi bien la programamtion des imprimantes. Pour cela, on fera appel au Basic, généralement fourni avec les machines, sous le nom de Basic, de Basica pour *IBM*, de GWBasic pour les compatibles, etc. Attention : *n'utilisez pas* Basica sur un compatible, car cela ne marche absolument pas.

Pour notre exemple, nous allons programmer une imprimante matricielle *Star* SR1Ø, compatible avec l'imprimante graphique *IBM*, afin qu'elle imprime les zéros barrés, donc Ø. C'est un fichier que l'auteur utilise d'office dans son système ; il est d'ailleurs appelé dans l' **AUTOEXEC.BAT** . Vous pourrez remplacer cette programmation par celle de l'écriture en gras, ou autre, à volonté (il suffit de repêcher dans le manuel de l'imprimante quels sont les codes de commande à appliquer).

Avec son XT, l'auteur appelle le Basica d'*IBM* :

```
C:\> BASICA
```

Le Basic apparaît aussitôt à l'écran et s'annonce :

```
The IBM Personal Computer Basic
Version A2.1Ø Copyright IBM Corp. 1981, 1982, 1983
6Ø455 Bytes free

Ok
```

Notez que c'est le Basic qui affichera des **" Ok "** , afin de vous rappeler éventuellement sa disponiblité.

Dès lors, frappez rigoureusement deux lignes ; la première est le code nullement secret (il a été relevé dans le manuel de l'imprimante) qui commande l'impression du zéro barré ; il s'écrit **CHR$(27)"\""1"** et est précédé d'une commande obligatoire **LPRINT** . La seconde ligne dit que c'est fini, qu'on revient au *système*, sous DOS :

```
1Ø LPRINT CHR$(27)"\""1"
2Ø SYSTEM
```

Dans votre cas, remplacez le **CHR$(27)"\""1"** par le code de votre demande, par exemple **CHR$(15)** tout simplement pour le condensé. Référez-vous à votre manuel.

Cela étant fait, on enregistre ce court programme, que l'auteur a appelé ØBARRE (utilisez toujours des noms significatifs) avec :

```
SAVE"ØBARRE
Ok
```

La sauvegarde achevée, l'écran ramène un OK. Revenez alors sous DOS en frappant :

```
Ok
SYSTEM
```

Dès lors et pour piloter l'imprimante et lui imposer les « Ø », il suffira d'appeler, sous DOS :

```
C:\DOS> BASICA ØBARRE
```

Mieux : si c'est un besoin constant, introduisez cet appel dans un fichier AUTOEXEC.BAT. Nous revenons à plusieurs reprises, dans diverses fiches dont certaines spécifiques, sur le rôle de l'AUTOEXEC.BAT, sur la façon d'en créer un ou de lui ajouter des lignes.

45

Pour travailler plus vite créez un « disque virtuel »

Problème

En travaillant avec vos programmes préférés, vous constatez que, souvent, le petit voyant indiquant que le disque dur est en activité s'allume. Simultanément, vous vous plaignez qu'il vous faille souvent attendre de longs instants avant d'obtenir ce que vous souhaitez sur l'écran, ou sur l'imprimante. Voici une méthode qui va vous permettre d'accélérer certaines opérations.

Solution

DISQUE VIRTUEL

Il vous faut un « disque virtuel » ! Mais qu'est-ce, au juste ? Les programmes travaillent normalement avec la mémoire centrale et avec les disques ; prenons le cas du disque dur, puisqu'il est des plus courants. Chaque fois qu'un programme doit aller chercher une donnée sur le disque dur, il perd environ un dixième de seconde. Un disque dur type, pour XT, par exemple, est caractérisé par ce qu'on appelle un *temps d'accès aléatoire*, de l'ordre de 80 ms (millièmes de seconde) ; il s'y ajoute les autres temps nécessaires à la lecture. Certains disque sont plus rapides (40 ms de temps d'accès, jusqu'à 15 ms pour les meilleurs), mais ils sont plus chers.

Notez que le temps d'accès n'est nullement le seul responsable. En effet, il faut lire les données sur le disque et les transmettre au microprocesseur ; or, la vitesse de transmission est limitée, et souvent inférieure à la vitesse de lecture. Aussi n'est-il pas rare qu'on freine volontairement la lecture à l'aide d'une astuce qui s'appelle l'*entrelacement* : au lieu de lire des données sur des secteurs du disque consécutifs, on les lit sur, par exemple, tous les quatre secteurs. Ce qui fait perdre en vitesse de lecture ou d'écriture. C'est vrai de la lecture comme de l'écriture, bien sûr.

Si les accès au disque se multiplient, ce sont des secondes ou des minutes qui sont consommées en pure perte. Pour pallier cet inconvénient, il suffit de réserver une portion de la mémoire centrale et de la baptiser *« disque »*. On va déclarer cette portion au système comme étant un disque supplémentaire, appelé un *disque virtuel*, et on va travailler avec lui comme si c'était, par exemple, le disque `D` (si vous disposez déjà du disque dur `C:` prenez toujours la lettre suivante, c'est plus simple !).

Le bénéfice est évident : un accès à la mémoire centrale demande entre 100 et 300 nanosecondes (ns) ; or, une nanoseconde est le millionnième d'un millième de seconde (10^{-9} s), c'est dire que le temps d'accès est d'office réduit par près d'un million ! En outre, les voies de liaison au microprocesseur (bus) acceptent, cette fois, la pleine vitesse.

Le disque virtuel offre toutefois un inconvénient : il vient au détriment de votre mémoire centrale (il se soustrait à la mémoire utilisable). Aussi sera-t-il souvent intéressant d'accroître la capacité de la mémoire centrale, selon l'une des méthodes présentée dans la fiche *« Pas assez de mémoire ? Voici comment accroître la capacité de la mémoire centrale »*. Un autre inconvénient subsiste, un réel danger cette fois :

Lorsqu'on éteint l'ordinateur, la mémoire centrale perd son contenu. Il faudra donc obligatoirement le sauvegarder sur disque au préalable.

VOLATILE

On dit de la mémoire centrale qu'elle est *volatile* (notez que cet inconvénient n'existe pas avec certains portatifs). Par conséquent, le processus va être le suivant : à la mise en service, on va transférer *sur le disque virtuel* (en mémoire centrale) les données utiles ; on travaillera puis, en fin de session, on les sauvegardera sur le disque réel, physique.

Pour créer un disque virtuel, plusieurs méthodes sont disponibles. Le plus simple consiste à faire appel au système d'exploitation (DOS), bien que perfidement, certains programmes conseillent de l'éviter et vous offrent une formule de rechange (par exemple, Windows).

Vérifiez que votre DOS comporte bien la séquence de création d'un disque virtuel, appelée :

VDISK.SYS

· `VDISK.SYS` , le plus souvent ; ou encore :

RAMDISK.SYS

· `RAMDISK.SYS` , car le *disque virtuel* est organisé dans les circuits de la mémoire centrale qu'on nomme RAM (prononcez : *« rame »*).

· `MDISK.SYS` ; etc. Consultez le manuel de votre DOS si vous ne trouvez aucun de ceux-là !

Il faut impérativement que cette séquence se trouve dans le répertoire principal du disque dur, si tel est le cas.

La présence de cette séquence vérifiée, il n'y a qu'une chose à faire, et elle est ultra-simple : indiquer au système qu'on veut un disque virtuel ! Pour ce faire, il suffit d'inclure une toute petite ligne dans *le premier fichier que lit l'ordinateur* à sa mise en service, nous avons nommé `CONFIG.SYS` . Il existe très certainement dans votre sys-

tème, car c'est généralement à lui qu'on déclare le pays d'utilisation. Pour afficher son contenu, on utilise la commande **TYPE** de façon classique ; voici ce que contenait celui de l'auteur (le vôtre sera différent, probablement) :

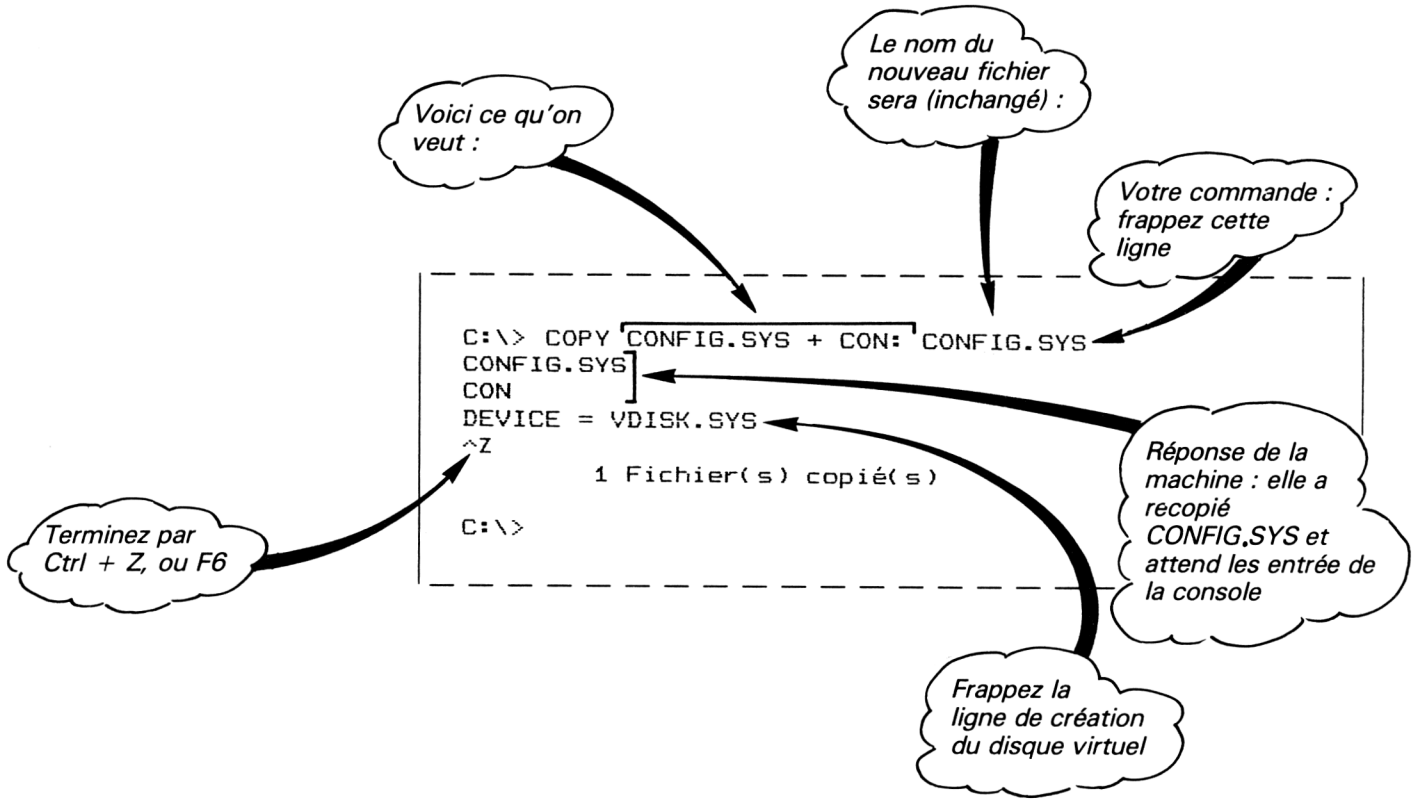
```
C:\> TYPE CONFIG.SYS
COUNTRY = 33
FILES   = 20
BUFFERS = 30

C:\>
```

Le mot **COUNTRY = 33** spécifie la France (c'est même son code téléphonique international). Négligez le reste, ici.

Or, il faut ajouter à cette liste la déclaration du disque virtuel, qui s'écrit, dans sa forme la plus dépouillée.

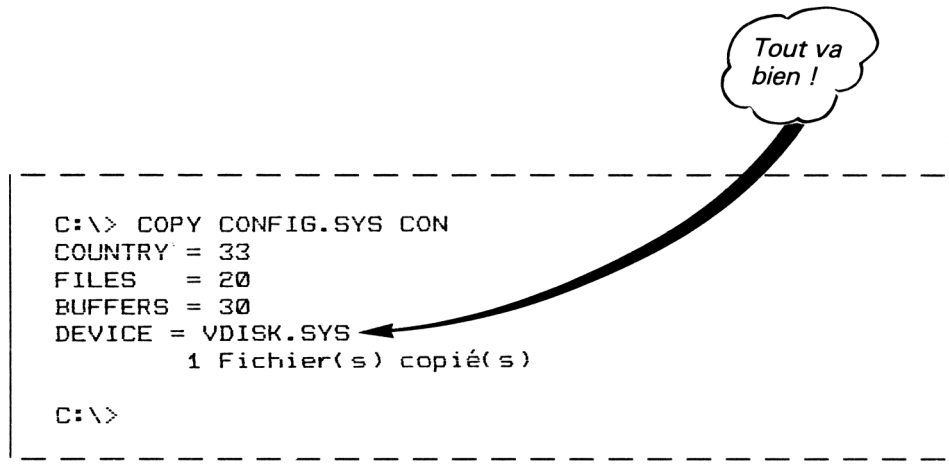
```
DEVICE = VDISK.SYS
```

Pour ajouter cette ligne, plusieurs méthodes sont disponibles. Passez par EDLIN ou par un traitement de texte, si vous le voulez (voyez notre fiche « Un court texte à créer ou à modifier ? Voyez EDLIN ! »). Sinon, la méthode la plus simple consiste à dire au système qu'on veut copier **CONFIG.SYS** *plus* quelque chose qu'on va frapper au clavier, sachant que pour la machine, le clavier, c'est *la console* et qu'on l'écrit **CON** en abrégé, avec un double-point (**CON:**) souvent facultatif. Respectez la séquence suivante :



Le *Ctrl* + *Z* final, qui affiche **^Z** à l'écran, est obtenu en frappant sur la touche F6 ; c'est le plus simple. N'oubliez pas de frapper *Entrée* à chaque fin de ligne. Le dernier enregistre le fichier **CONFIG.SYS** modifié, et le système affiche son message d'exécution réussie.

Pour revoir le contenu du nouveau **CONFIG.SYS** , on peut se servir du même **COPY** sur la console, mais en inversant la position de **CON:** (si vous préférez, faites un **TYPE** , comme précédemment) :



L'opération est terminée. Pour que l'ordinateur prenne en compte le disque virtuel, il faut le remettre en service, ou le ré-initialiser avec *Ctrl* + *Alt* + *Del* ; vous verrez alors s'afficher à l'écran un message témoignant de la création du disque virtuel (auquel le DOS a tout seul, comme un grand, attribué la lettre **D**) :

```
VDISK Version 3.30 disque virtuel D:
   Taille mémoire tampon ajustée
   Taille secteur ajustée
   Entrées de répertoires ajustées
   Taille mémoire tampon :        64 KB
   Taille secteur :              128
   Entrées de répertoire :        64
```

D'office, et parce qu'on n'a rien spécifié d'autre, la capacité du disque virtuel est de 64 K octets (64 KB, avec B pour *Byte*, octet, en anglais) ; chaque « secteur » représente 128 octets, et vous pourrez enregistrer 64 noms de fichiers dans le répertoire.

Attention : ces 64 Ko de disque virtuel viennent *en moins* sur vos 640 Ko disponibles. Si vous disposez d'une mémoire supplémentaire, *étendue* (normes EMS ou EEMS : voyez notre fiche *« Pas assez de mémoire ? Voici comment accroître la capacité de la mémoire centrale »)*, vous pourrez loger ces 64 Ko dans cette mémoire étendue et conserver intacts vos 640 Ko. Pour cela, il faut associer une « clé » à la commande ; cette clé, c'est l'ensemble **/E** . Remplacer le **DEVICE = VDISK.SYS** par :

```
DEVICE  = VDISK.SYS /E
```

Votre **CONFIG.SYS** apparaîtra alors ainsi :

```
C:\> TYPE CONFIG.SYS
COUNTRY = 33
FILES   = 20
BUFFERS = 30
DEVICE = VDISK.SYS  /E

C:\>
```

Si vous souhaitez, non pas 64 Ko de disque virtuel mais, par exemple, 256 (et ce, en restant *dans* les 640 Ko), il suffirait de préciser avec, cette fois :

```
C:\>    DEVICE = VDISK.SYS 256
```

Pour loger cela *au-dessus* des 640 Ko, si vous êtes équipé d'une carte étendue, il faudra ajouter **/E** .

```
DEVICE = VDISK.SYS 256  /E
```

On peut encore modifier la longueur d'un secteur, et le nombre d'entrées dans le répertoire ; abstenez-vous de le faire en un premier temps, sauf obligation. Par exemple, la ligne suivante crée un disque virtuel de 256 Ko, avec des secteurs de 512 octets et 120 entrées dans le répertoire, le tout logé dans la mémoire d'extension :

```
DEVICE = VDISK.SYS 256 , 512 , 120  /E
```

Si vous tentez d'utiliser la clé **/E** sans disposer de mémoire d'extension, le système vous le signalera obligeamment :

```
VDISK Version 3.30 disque virtuel D:
   Taille secteur ajustée
   Entrées de répertoires ajustées
   Taille de transfert ajustée
   VDISK pas installé - mémoire insuffisante
```

Le disque virtuel créé, utilisez-le tout comme s'il s'agissait d'un disque quelconque. Mais vous apprécierez la rapiditié et, partant, le confort qu'ils vous procurera pour des applications exigeantes.

Notez encore que vous pouvez inclure le nom du disque virtuel dans un **PATH** . Dans ce cas, il est de bon ton de le placer en tête de liste, car c'est par là que le DOS commence sa lecture pour rechercher des fichiers. Profitez-en pour ranger sur le disque virtuel des programmes d'usage fréquent ; par exemple, supposons que vous faites souvent appel à la séquence **CHKDSK** de test de disque du DOS ; recopiez-le du disque dur habituel (si tel est le cas, ou sinon, de sa disquette), sur le disque virtuel en incluant dans l' **AUTOEXEC.BAT** :

```
C:\> PATH D:\;C:\;C:\DOS
     COPY C:\DOS\CHKDSK.COM D:\
```

Vous y aurez ainsi accès très rapidement. Rappelez-vous que **DISKCOPY** ne fonctionne pas avec un disque virtuel.

46

Pas assez de mémoire ? Voici comment accroître la capacité de la mémoire centrale

Problème

On veut utiliser un, ou plusieurs programmes, mais un message inquiétant apparaît à l'écran : « Pas assez de mémoire ». Que faire ?

Solution

Parfois, vous saurez que vous ne disposez pas de suffisamment de mémoire centrale car, lors du chargement d'un programme, l'écran affiche le terrible message : *« Pas assez de mémoire ! »*. Le programme ne sera pas chargé, et vous ne pourrez pas l'exécuter. D'autres fois, le manque de mémoire n'entravera pas le chargement des programmes mais se traduira par :

· Une limitation dans leur capacité de traitement (importance d'un tableau avec les tableurs, d'un texte avec des traitements de texte, etc.).
· Un ralentissement sensible dans la vitesse d'exécution. Le programme se voit contraint de ranger bien des choses sur disque, et organise un ballet permanent mémoire centrale à disque, et vice-versa, ce qui exige du temps.

La solution consiste alors, dans tous ces cas, à accroître la capacité de votre mémoire centrale. Sachez que la portion de mémoire centrale réservée à l'utilisateur (et à ses programmes) est, au maximum, de 640 K octets (caractères) avec les PC/XT/AT. Cette limite est imposée par le microprocesseur et/ou par le système d'exploitation DOS. En fait, ceux-ci savent gérer jusqu'à 1 Mo de mémoire centrale, mais le système, l'écran, etc., s'en réservent 384 Ko ; il vous en reste donc 640 Ko dans le meilleur des cas.

Dans le meilleur des cas, car il n'est pas évident que votre machine vous ait été livrée avec ses « 640 Ko utilisateur ». A l'origine, on ne vous livrait que 64 Ko ; puis, c'est passé à 128 Ko, puis 256 Ko, puis 320 Ko..., pour atteindre, actuellement et le plus souvent, 520 Ko ou 640 Ko.

Afin de vérifier de combien de mémoire vous disposez, il suffit d'observer l'écran de votre micro à sa mise sous tension. Il exécute alors un auto-test de démarrage et vérifie que ses mémoires fonctionnent bien, en affichant leur capacité.

Si vous ne disposez pas de 640 Ko, vous pourrez tout d'abord commencer par compléter votre mémoire centrale, selon deux méthodes :

1) — La « carte-mère » de votre machine est équipée de supports destinés à recevoir des circuits intégrés mémoires supplémentaires. Vérifiez-en le type, achetez ces circuits, et introduisez-les dans leurs supports. *Attention : ne tordez pas leurs pattes !* Il s'agit là d'un petit exercice demandant une certaine dextérité.

2) — Vous utilisez une « carte d'extension mémoire », achetée dans le commerce, tout équipée de ses circuits. Introduisez-la dans l'un des connecteurs d'extension de votre machine.

Dans ces deux cas, n'oubliez pas de repositionner les éventuels interrupteurs présents sur la carte-mère, afin de signifier au système à quelle capacité vous avez porté la mémoire centrale.

Pour cela, référez-vous au manuel d'utilisation de votre machine.
Fort bien : mais si 640 Ko se révèlent un peu « justes » ? Certains programmes sont très gourmands en mémoire, et pénalisent l'utilisateur, s'ils n'en trouvent pas assez, en lui imposant des attentes jugées toujours excessives. C'est le cas des grands tableaux, sous Lotus 1-2-3, par exemple, qu'il faut recalculer. C'est le cas avec le programme intégrateur Windows et tout ce qu'il gère, avec le programme de publication assitée Pagemaker (ou d'autres), avec certains traitements de texte qui ne savent travailler qu'avec la mémoire centrale (celle-ci limite la longueur des documents utilisables), etc.

Dès lors, il ne subsite plus qu'une méthode, *outrepasser le 1 Mo* de mémoire centrale *totale*, car, fait important, certains programmes sauront l'utiliser. Par exemple, Lotus 1-2-3, Symphony, Windows, Paradox, Q & R (Q & A, aux Etats-Unis), Manuscript, etc. Pour ce faire, il faut introduire dans l'unité centrale une *carte d'extension* mémoire, avec la capacité supplémentaire requise ; elle pourra aller jusqu'à plusieurs méga-octets. Deux techniques sont appliquées, selon les cartes :

EXPANDED MEMORY SPECIFICATION

· *L'Expanded Memory Specification*, ou EMS. Elle a été développée par les sociétés *Lotus, Intel* et *Microsoft*, raison pour laquelle on traite de concept *LIM*. Les programmes d'application la gèrent par tranches de 16 Ko, appelées « pages ».

EMS

· *L'Enhanced Expanded Memory Specification*, ou EEMS, développée par *Ast Research*. Les pages sont gérées sur 64 Ko.

La plupart des programmes n'exploiteront cette mémoire supplémentaire que pour *leurs données*. Vous pourrez également vous en servir pour créer des *disques virtuels*. Voyez comment avec la recette consacrée à se sujet.

ENHANCED EXPANDED MEMORY SPECIFICATION

Une confusion reste possible dans le mode d'expression : on dira de ces extensions qu'*elles brisent la barrière des 640 Ko*, ou *la barrière de 1 Mo*, selon qu'on prend en considération la seule capacité de mémoire exploitable par les programmes de l'utilisateur, ou sa totalité. Ce résultat reste le même ! Notons aussi que la nouvelle norme EMS, version 4, regroupe l'EMS et l'EEMS.

EEMS

A l'avenir, grâce à l'usage des microprocesseurs 80286 et 80386, associés au système d'exploitation OS/2, vous pourrez disposez directement de 16 Mo de mémoire centrale.

47

Répertoires et chemins d'accès : à quoi ça sert et comment les organiser

Problème

Pour le débutant, les notions de répertoires, sous-répertoires, chemin d'accès, « PATH », sont déroutantes. De quoi s'agit-il réellement et comment faut-il s'organiser ?

Solution

REPERTOIRE RACINE

REPERTOIRE PRINCIPAL

Un disque, surtout un disque dur, est capable de contenir des quantités de choses différentes. Par exemple, des programmes divers (traitement de texte, gestion de fichiers, gestion tout court, etc.) ainsi que des fichiers (le courrier que vous réalisez, les divers tableaux de votre gestion, des graphiques, etc.). Si le nombre de titres (programmes et fichiers) est restreint, par exemple une dizaine, le *« catalogue général »* du disque, qu'on appelle *le répertoire principal*, ou encore le répertoire racine, pourra en dresser la liste et elle restera lisible.

Or, les simples programmes du système d'exploitation, du DOS, sont déjà au nombre de plusieurs dizaines ! Si on les mélange avec le reste, et que ce reste tend à appliquer ce *« Croissez et multipliez ! »* qui n'était certainement pas destiné à la micro, la *« table des matières »* du disque (toujours appelée *répertoire principal*) va représenter plusieurs écran successifs, pleins à ras bord, et dans lesquels on aura la plus grande peine à détecter « le » fichier qu'on recherchait.

Autant procéder, alors, avec ordre et méthode. Comme dans toute bonne bibliothèque, on va classer toutes les séquences selon leurs rubriques. Au lieu des *Romans, Essais, Science-fiction, Nouvelles ou Poésie*, d'une librairie, on aura *Courrier* (par exemple...) pour le traitement de texte et les lettres rédigées, *DOS* pour les programmes du DOS, *GRAPHIQUE* pour le logiciel graphique et tout ce qu'on aura développé avec lui, etc. On va appeler ces répertoires des *sous-répertoires* ; puis, on réunira leurs noms dans le *répertoire principal.*

Si nécessaire, c'est-à-dire, si vous multipliez encore le nombre des fichiers, vous pourrez établir des sous-sous-répertoires ; par exemple, et pour reprendre la comparaison avec une bibliothèque, on trouvera dans l'index général (le *répertoire principal*) le titre *Romans*, puis dans le répertoire *Romans*, les sous-titres : *Classiques, Aventures,...* ceux-ci mènent enfin aux pages ou sont listés les *Romans français*, puis les *Romans étrangers*, etc. Transposée aux répertoires, la même structure mène aux :

SOUS-REPERTOIRES

· *Répertoire principal :* ne lui confiez que les titres, avec le minimum des programmes nécessaires à l'initialisation de la machine (au moins, le CONFIG.SYS, l'AUTOEXEC.BAT, et quelques petites choses encore).
· *Des sous-répertoires*, par grands thèmes d'applications.
· *Eventuellement, des sous-sous-répertoires*, bien pratiques pour ranger les documents qu'on crée. On les retrouve plus facilement, et l'on fait plus aisément « le ménage ».

SOUS-SOUS REPERTOIRES

Oui, mais voilà : comment atteindre le fichier VENTES.MAI, situé dans le sous-sous-répertoire VENTES.AN, lui-même situé dans le répertoire GESTION ? C'est simple : en spécifiant le trajet GESTION, VENTES, VENTES.MAI. Mais au lieu d'utiliser des virgules, on place des barres obliques inversées, obtenues en frappant les touches *Alt* + *Ctrl* (enfoncées) et la touche portant la barre oblique inversée, parfois sur son flanc. Ce qui donne :

```
\GESTION\VENTES\VENTES.MAI
```

La première barre oblique inversée spécifie *le répertoire principal*, dans lequel on trouve le titre GESTION. Une seconde barre oblique marque le premier sous-répertoire, VENTES, dans lequel on trouvera enfin (séparé par la seconde barre oblique : il faut bien établir une différence) le nom du fichier VENTES.MAI.

L'auteur, qui spécifie toujours avec les caractères d'appel du DOS le nom du répertoire actif et le *chemin qui y mène* (la séquence ci-dessus) — voyez la fiche particulière consacrée à ce thème — fait aussi apparaître à l'écran :

```
C:\GESTION\VENTES>
```

Ici, c'est VENTES le nom du sous-répertoire « actif », en service.

CD

Pour changer de répertoire, on ordonne CD, pour « *Change Directory* », en anglais. Afin de revenir au *répertoire principal*, on frapperait simplement :

CHANGE DIRECTORY

```
C:\GESTION\VENTES> CD \

C:\>
```

puisque la *première barre oblique inversée* représente *le répertoire principal.*

Le répertoire principal est encore appelée « répertoire racine », car il est à l'origine de tous les autres, ou « répertoire maître ».

Revenir au sous-répertoire VENTES s'écrirait, toujours avec un CD :

```
C:\> CD \GESTION\VENTES

C:\GESTION\VENTES>
```

Ainsi, le chemin d'accès à un fichier est l'enfilade des répertoires qui y mènent.

PATH

Chemin d'accès se dit PATH, en anglais. Il s'agit là d'une commande qu'on pourra inclure dans l'AUTOEXEC.BAT afin d'indiquer au système dans quels sous-répertoires successifs il devra rechercher des fichiers qu'il n'a pas trouvés dans le répertoire actif en cours.

Nous en traitons, dans une autre fiche. C'est une bonne habitude, *dès le début*, d'organiser le contenu du disque dur avec la plus grande rigueur. Notez encore ce fait important :

Grâce à l'organisation en sous-répertoires, des noms de fichiers *identiques* peuvent désigner des fichiers *différents* s'ils sont placés dans des sous-répertoires *différents*.

Par exemple, les fichiers CLIENTS seront différents avec :

```
\MAGASIN1\CLIENTS
\MAGASIN2\CLIENTS
```

Cela, parce que leur chemin d'accès est, lui aussi, différent.

48

Quelques trucs relatifs aux sous-répertoires

Problème

On va aborder, ici, plusieurs problèmes de navigation et de traitement des sous-répertoires, en vous présentant des méthodes pratiques pour accélérer les manipulations.

1. EFFACER LE CONTENU D'UN SOUS-RÉPERTOIRE

EFFACER LE CONTENU D'UN SOUS-REPERTOIRE

Pour effacer tous les fichiers d'un sous-répertoire (votre disque est plein, et vous êtes contraint d'élaguer !), vous pouvez vous placer *dans* ce sous-répertoire, appelé DEFUNT, par exemple, et commander :

```
C:\DEFUNT> DEL *.*
Etes-vous sûr (O/N) ?O

C:\DEFUNT>
```

Une autre méthode consiste à passer au niveau au-dessus, et à commander :

```
C:\DEFUNT> CD ..

C:\> DEL \DEFUNT\*.*
Etes-vous sûr (O/N) ?
```

On peut encore faire plus court avec, tout simplement :

```
C:\> DEL DEFUNT
Etes-vous sûr (O/N) ?
```

Il est inutile de spécifier ***.*** . Vous serez aussitôt en place, ensuite pour commander un **RD DEFUNT** . Notez que dans tous les cas, le DOS exigera une confirmation avant d'effacer totalement tous les fichiers.

2. REMONTER DE PLUSIEURS NIVEAUX DE SOUS-RÉPERTOIRES

Supposons que l'on se trouve dans le sous-(sous-sous...)-répertoire, OMEGA de la chaîne :

```
C:\> C:\ALPHA\BETA\GAMMA\DELTA\EPSILON\OMEGA
```

et qu'on veuille remonter à GAMMA. Ne retapez pas la chaîne, commencez par faire :

```
CD ..
```

suivie d'une *Entrée.* Puis, faites des F3 (frappe de la touche de fonction F3) pour répéter la commande, et des *Entrées,* jusqu'à ce que vous soyez remonté au bon niveau. Pour une remontée totale (nous vous le montrons pour la beauté de la chose, car il serait plus simple, dans ce cas, de faire directement, **" CD \ "** , voyez aussitôt après), on obtiendrait :

```
C:\ALPHA\BETA\GAMMA\EPSILON\OMEGA> CD ..

C:\ALPHA\BETA\GAMMA\EPSILON> CD ..

C:\ALPHA\BETA\GAMMA> CD ..

C:\ALPHA\BETA> CD ..

C:\ALPHA> CD ..

C:\>
```

Arrêtez-vous en chemin où bon vous semble.

3. REVENIR AU RÉPERTOIRE PRINCIPAL

La commande est connue, mais certains débutants peuvent l'ignorer. Rappelons donc que, pour remonter directement au *répertoire racine* (répertoire principal), il suffit de faire :

```
C:\> CD \
```

puisque la première barre oblique inversée est son symbole.

4. PASSER À UN SOUS-RÉPERTOIRE DE MÊME NIVEAU

Supposons qu'on soit dans :

```
C:\>  C:\ALPHA\BETA\STOCKS
```

et qu'on veuille passer à :

```
C:\>  C:\ALPHA\BETA\CLIENTS
```

Il suffira de frapper :

```
C:\ALPHA\BETA\STOCKS> CD ..\CLIENTS

C:\ALPHA\BETA\CLIENTS>
```

puisque les deux points alignés (..) sont le symbole du répertoire père, de niveau immédiatement supérieur.

5. SAUTER ALLÈGREMENT D'UNE BRANCHE À L'AUTRE

Supposons qu'on se trouve dans ALPHA, avec le programme CLIENTS :

```
C:\> C:\ALPHA\CLIENTS
```

et qu'on veuille passer à :

```
C:\BETA\DELTA\GAMMA\EPSILON\OMEGA\STOCKS
```

Plutôt que de retaper cette liste excécrable, il vaudrait mieux pouvoir faire directement :

```
C:\> STOCKS
```

et changer automatiquement de *branche*, dans cette arborescence. C'est possible : nous vous conseillons même d'établir les conditions de ce transfert automatique :

1) — Il faut créer un fichier **.BAT** , par exemple **STOCKS.BAT** tel que :

```
C:\> COPY CON STOCKS.BAT
CD \BETA\DELTA\GAMMA\EPSILON\OMEGA
STOCKS
^Z
        1 Fichier(s) copié(s)

C:\>
```

2) — Rangez ce fichier dans un répertoire spécifique, réservé aux fichiers **.BAT** , ou encore dans le répertoire principal.

3) — Incluez son chemin dans l'**AUTOEXEC.BAT** , avec un :

```
PATH C:\
```

si c'est le répertoire principal qui le contient. Vous pouvez en faire autant avec d'autres fichiers. Voyez, à ce propos, les fiches que nous avons consacrés aux chemins de rechange (avec **PATH**), et à la réalisation d'un fichier **.BAT** , directement au clavier (avec **COPY CON**). Dès lors, la frappe de STOCKS, où que l'on se trouve, mènera directement à ce fichier.

49

Comment afficher le nom du sous-répertoire actif avec les caractères d'appel du DOS

Problème

On a créé, sur le disque dur, plusieurs sous-répertoires. Or, et tant qu'on reste sous DOS, avec les caractères d'appel `C>` *, on ne sait absolument pas dans quel répertoire actif on se trouve. C'est source de confusion et d'erreurs. Comment résoudre ce problème ?*

Solution

Il suffit d'afficher, avec les caractères d'appel, la chaîne des sous-répertoires jusqu'au répertoire actif. Pour cela, il nous faut revenir à la définition des caractères d'appel.

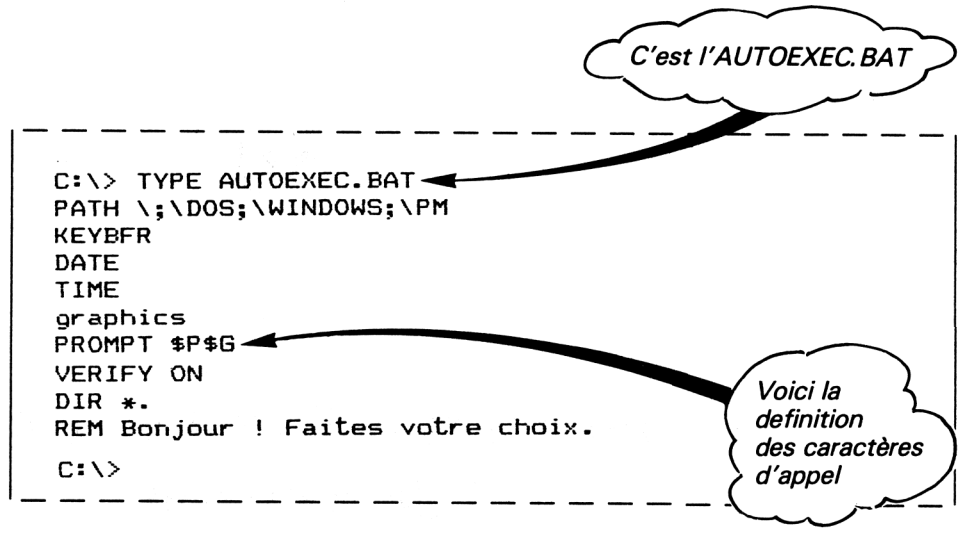
Par défaut et à la réception d'une machine, les caractères d'appel apparaissant à l'écran sont :

`A>` si le disque A est actif,
`C>` si c'est le disque dur C qui est actif.

Ou toute autre lettre (B, D...) si un autre disque est actif. Ces caractères d'appel, le `PROMPT` , pour les Américains, peuvent être redéfinis par une commande de même nom *qu'on incluera dans l'* `AUTOEXEC.BAT` . On y écrira le code suivant, par exemple :

```
C:\> PROMPT $P$G
```

Ces codes vont afficher la chaîne et le sous-répertoire actif, suivis par le symbole >. Laissez même un espace, à la fin, avant de frapper l'*Entrée*, pour la beauté de la présentation. Vous retrouverez cette définition dans l' **AUTOEXEC.BAT** de l'auteur :



C'est ce qui explique que les caractères d'appel ne soient plus **C>** mais **C:\>** pour le *répertoire principal*, et par exemple, pour les sous-répertoire VENTES :

```
C:\GESTION\VENTES>
```

C'est tellement simple à obtenir qu'il serait stupide de se passer d'une information aussi précieuse. Puisque nous y sommes, ajoutons que PROMPT permet bien d'autres choses intéressantes, ou fantaisistes ; par exemple, essayez, directement sous DOS (et non dans votre AUTOEXEC.BAT) :

```
C:\> PROMPT $L$N$G $L$N$G $_    $B $_ $L$Q$Q$Q$G $_
```

Si vous avez bien compté les espaces (d'abord 3, puis 1), vous devriez obtenir :

```
<C> <C>
   !
 <===>
```

Selon les critères actuels, on peut considérer que c'est aussi une œuvre d'art.

50

Comment spécifier des répertoires de rechange

Problème

On travaille avec des fichiers répartis dans divers sous-répertoires. Or, le DOS ne semble connaître que le seul répertoire actif. Comment lui expliquer que, s'il n'y trouve pas le fichier demandé, il doit aller le chercher dans un autre répertoire ?

Solution

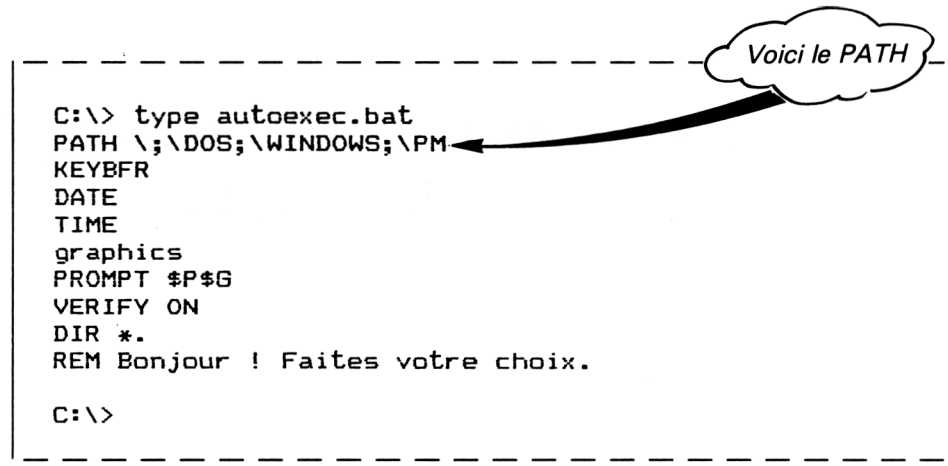
Il s'agit là d'un problème des plus fréquents. L'exemple le plus élémentaire est le suivant : on se trouve, en `C` , dans le répertoire principal, et l'on veut formater une disquette en `A` . Or, la séquence `FORMAT` du DOS se trouve dans le sous-répertoire « DOS » du disque dur. Est-il vraiment nécessaire de changer, d'abord, de répertoire ? Un cas plus complexe surgit lorsqu'on travaille avec un programme d'application et qu'on veut faire appel à une commande située dans un autre sous-répertoire : le système s'y refusera. A moins que l'on n'ait, au préalable, fourni au système des directives précises spéficiant dans quels autres sous-répertoires il doit aller chercher la séquence absente du sous-répertoire actif, courant.

Cette déclaration de répertoires de rechanges, alternatifs, avec l'énoncé de leurs chemins d'accès, se fera plus commodément si on l'introduit dans le fichier `AUTOEXEC.BAT` . Les règles sont les suivantes :

- On écrit le mot `PATH` (*chemin*, en anglais)
- On indique le chemin complet menant au sous-répertoire désiré, à partir du répertoire principal, spécifié par une première barre oblique inversée.
- Eventuellement (et en cas de doute !), on notera également le nom du disque visé au préalable, par exemple `C:`
- On peut indiquer d'autres répertoires, avec leur chemin d'accès, *sur la même ligne*, en respectant ces règles :

- **On sépare deux voies par un point-virgule.**
- **Attention : n'introduisez pas d'espace !**

Avec un espace, le système pourrait supposer que la déclaration est terminée.
Voici un exemple de ce que cela peut donner (cas de l'auteur), dans un extrait d' **AUTOEXEC.BAT** :



Le **PATH** de la première ligne de cet AUTOEXEC.BAT ordonne au système de rechercher un fichier, d'abord et implicitement, dans le répertoire actif, sinon et dans l'ordre : dans le répertoire principal, racine (\ ) ; sinon, dans le sous-répertoire DOS (**\DOS**) qui regroupe les commandes externes du DOS ; sinon, dans le sous-répertoire WINDOWS (avec \WINDOWS) ; ou enfin, dans le sous-répertoire PM (c'est le programme de publication assistée PageMaker). Comme tout se passe sur C, on ne l'a pas spécifié ; on aurait pu le faire en écrivant, par exemple :

```
C:\>    PATH C:\;C:\DOS;C:\WINDOWS;C:\PM
```

Si, au cours de votre travail, vous souhaitez connaître les chemins et sous-répertoires alternatifs, vous pouvez le demander au DOS en frappant simplement **PATH** . Dans notre cas, il répondrait :

```
C:\> PATH
PATH=\;\DOS;\WINDOWS;\PM

C:\>
```

Rappelez-vous que `PATH` n'est utile que pour des fichiers de programme, avec l'extention `.EXE` , ou `.COM` , ou `.BAT` .

Pour des fichiers de données et avec les seuls DOS à partir du 3.2, il faudrait employer **`APPEND`** (voyez la fiche que nous lui avons consacré). Songez encore à utiliser des fichiers **`.BAT`** , situés dans le répertoire principal, pour mener *directement* à n'importe quel fichier de programme situé dans n'importe quel sous-répertoire, et ce, à partir de n'importe quel sous-répertoire servant de point d'appel (voyez la fiche qui traite de ce thème).

51

Comment appeler un programme directement, où qu'il se trouve sur le disque

Problème

Le disque dur est organisé en répertoire principal, sous-répertoires, sous-sous-répertoires, etc. On veut pouvoir appeler un programme se trouvant dans ***n'importe quel*** *sous-répertoire (quel que soit son niveau), et ce, à partir du sous-répertoire absolument quelconque où l'on se trouve actuellement.*

Solution

Nous vous suggérons une méthode simple et efficace permettant de naviguer à volonté parmi les répertoires, en les ignorant superbement. Elle est idéale pour les débutants, pour tous ceux qui ne veulent pas s'embarasser et retenir l'organisation hiérarchique de leur disque dur, ni utiliser des « outils » qui, sous prétexte de leur simplifier les manipulations, commencent par les compliquer.

En voici un exemple simple, d'où l'on a exclu les niveaux successifs des sous-répertoires afin de rendre la démonstration plus évidente. On se trouve dans le sous-répertoire QR, où l'on travaille sur une gestion de fichiers. Cela terminé, on veut passer au traitement de texte WORD, lequel se trouve dans le sous-répertoire WORD. Ce que nous vous proposons de faire, c'est d'appeler *directement* WORD à partir de QR, et cela, soulignons-le, bien qu'ils soient situés dans des sous-répertoires différents :

```
C:\QR> WORD
```

On ignore donc les sous-répertoires, on s'épargne leurs frappes ainsi que celle des barres obliques inversées, qui imposent toujours une certaine gymnastique intellectuelle et digitale.

Afin d'aboutir à ce résultat, il faut deux choses :

1) — D'abord, introduire dans la séquence AUTOEXEC.BAT qui se trouve déjà sur le disque une ligne spécifiant que le système devra d'office se reporter au répertoire principal s'il ne trouve pas ce qu'on lui demande dans le répertoire courant.

2) — Puis, créer un programme **.BAT** , rangé *dans le répertoire principal*, pour simplifier, qui dirigera le système vers le sous-répertoire où se trouve le programme appelé, et le lancera.

Voici comment procéder :

· Dans l'AUTOEXEC.BAT, il faut introduire la ligne :

```
PATH \
```

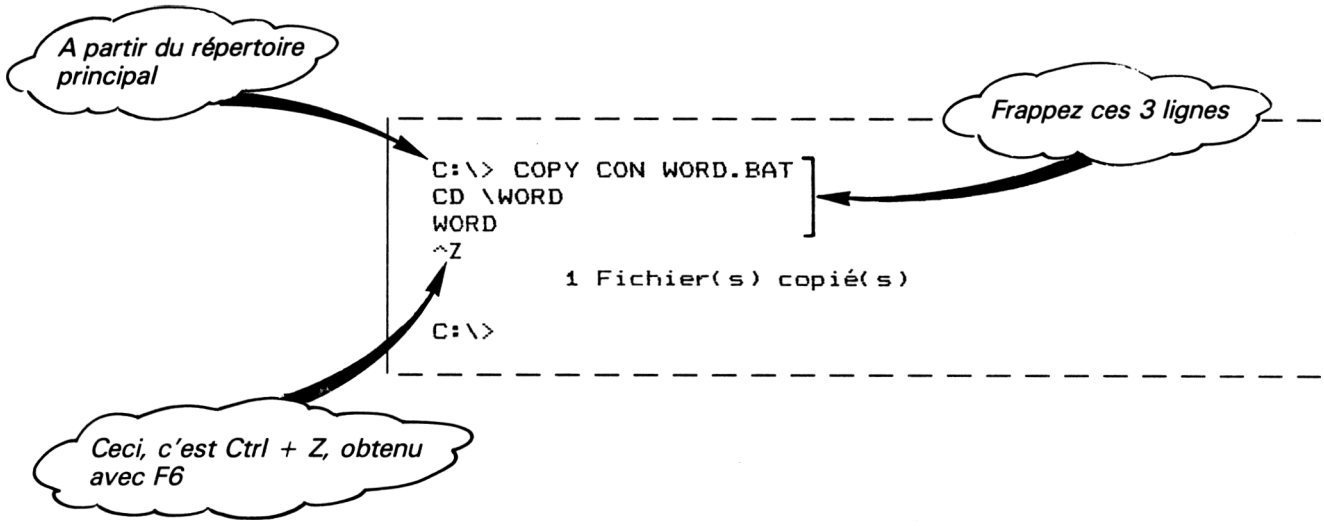
Vérifiez, au préalable, qu'elle ne s'y trouve pas déjà, en plus développé, même. Reportez-vous à notre recette « *Comment spécifier des répertoires de rechange* » si vous ne savez pas comment vous y prendre. Si, réellement, vous deviez créer ici cette ligne, il vous faudrait ensuite ré-initialiser l'ordinateur avec un *Ctrl* + *Alt* + *Del* pour la rendre active. L'auteur, lui, disposait déjà, au tout début de son AUTOEXEC.BAT, de :

```
C:\> TYPE AUTOEXEC.BAT
PATH \;\DOS;\WINDOWS;\PM
KEYBFR
DATE
TIME
graphics
PROMPT $P$G
CD \SOURIS
MOUSE
SET TEMP=C:\PM
CD \IMPRIM
HARD.COM
CD \
VERIFY ON
DIR *.
REM Bonjour ! Faites votre choix.

C:\>
```

· Puis, on crée *une fois pour toutes* la séquence suivante, obligatoirement dans le répertoire principal et sous le nom de **WORD.BAT** (dans notre cas : dans le vôtre, faites **TABLEUR.BAT** ou **VENTES.BAT** , etc., seule l'extension **.BAT** étant obligatoire ; le nom principal sera celui sous lequel vous appellerez votre programme :

```
CD \WORD
WORD
```

Afin de composer cette séquence, le plus simple consiste à passer dans le répertoire principal, puis à créer un fichier directement au clavier en respectant la séquence suivante :



CTRL + Z

Le Ctrl + Z final, qui indique au système que le fichier est terminé, est obtenu très simplement en frappant sur F6. Faites des *Entrées* après chaque ligne. Le message « *1 Fichier(s) copié(s)* » est affiché par le système.

Dès lors, partez dans un sous-répertoire quelconque, et frappez *uniquement* WORD : le programme sera appelé et viendra à l'écran.

Un conseil : procédez de la même façon avec vos programmes importants, vous vous épargnerez des frappes et des réflexions inutiles.

52

Comment afficher la seule liste des sous-répertoires

Problème

Le répertoire principal demande plusieurs écrans pour être affiché dans sa totalité. Or, seule la liste des noms des sous-répertoires qu'il contient se révèle intéressante, lors du travail quotidien. Peut-on sélectionner et n'afficher que cette seule liste ?

Solution

Oui, et cette commande est ultra simple puisqu'elle se borne à un :

```
DIR *.
```

```
C:\> DIR *.
```

Vous avez probablement remarqué que, lorsqu'on affiche le contenu du répertoire principal avec un DIR, les noms des sous-répertoires apparaissent *sans* extension. Or, si vous avez suivi les conseils que nous vous prodiguons dans ce livre, vous ne conservez dans votre répertoire principal que :

1) — *Les noms des séquences essentielles*, vitales et inévitables nécessaires à l'initialisation de la machine. Il s'agit de programmes *dotés d'une extension* qui est soit `.COM` , soit `.EXE` , soit encore `.BAT` ou `.SYS` .

2) — *Les sous-répertoires* qui, eux, ne possèdent pas d'extension.

Par conséquent, la frappe de **DIR *.** n'affichera que les nom des sous-programmes, ce qui se traduit par (cas de l'auteur, lors de la rédaction de ce livre) :

```
C:\> DIR *.

 Volume dans unité C est LILEN
 Répertoire de  C:\

VENTE         <DIR>       1/07/87   3:43
DOS           <DIR>      17/04/87   9:22
TRANSFER      <DIR>       1/01/80   0:43
COPY          <DIR>      17/07/87   1:12
IMPRIM        <DIR>      17/04/87   9:35
SOURIS        <DIR>      17/04/87   9:36
ADDICIEL      <DIR>      17/04/87   9:38
QR            <DIR>      17/04/87   9:39
TB            <DIR>       1/01/80   0:01
WINDOWS       <DIR>       1/01/80   2:25
PCTOOLS       <DIR>      17/07/87   1:14
LIVREPAO      <DIR>      17/08/87   9:54
MP            <DIR>      22/07/87   8:54
XTREE         <DIR>       1/08/87   2:12
MSTOOLS       <DIR>      27/07/87   9:02
WORD          <DIR>      29/07/87   0:06
        16 Fichier(s)
   1163264 octets disponibles

C:\>
```

Seuls, les sous-répertoires sont listés. Pour information, voici la liste des autres titres que contient le répertoire principal :

COMMAND.COM
COUNTRY.SYS
KEYBFR.COM
CONFIG.SYS
PMTUTOR.BAT
VDISK.SYS
WORD.BAT
AUTOEXEC.BAT

Ce qui, au total, excédait la capacité de 25 lignes d'un écran avec un affichage du répertoire complet.

Compléments

Les noms des touches du clavier

TOUCHES DES CLAVIERS

Les noms des touches des claviers (84 et 102 touches) ont évolué et, de leur anglais d'origine, ont enfin été francisés. Voici les principales correspondances.

Claviers PC non francisés	Clavier AT mod. 1 & 2	Clavier 102 touches
Home	↖	↖
End	Fin	Fin
PgUp	⇞	⇞
PgDn	⇟	⇟
Ins	Ins	Inser
Del	Annul	Suppr
Break	Attn	Attn
PrtSc	Imp Ec	Impr écran
Num Lock	Verr num	Verr num
Scroll Lock	Arrêt défil	Arrêt défil
Caps Lock	Maj	⇪
Esc	Esc	Échap

Le code ASCII

Le code ASCII (« Americain Standard Code for Information Interchange ») code tous les caractères :

CODE ASCII

- *De base, de 0 à 127 en décimal ;*
- *Etendus, de 128 à 255.*

Nous vous le présentons ici en deux tableaux, pour plus de commodité.

Poids	Caractère ou code	
000	(nul)	NUL
001	☺	SOH
002	●	STX
003	♥	ETX
004	♦	EOT
005	♣	ENQ
006	♠	ACK
007	(''bip'')	BEL
008	(Espace arrière)	BS
009	(Tabulation)	HT
010	(Ligne suivante)	LF
011	(Home)	VT
012	(Page suivante)	FF
013	(Retour chariot)	CR
014	♫	SO
015	☼	SI
016	►	DLE
017	◄	DC1
018	↕	DC2
019	!!	DC3
020	π	DC4
021	§	NAK
022	▬	SYN
023	↨	ETB
024	↑	CAN
025	↓	EM
026	→	SUB
027	←	ESC
028	(Curseur à droite)	FS
029	(Curseur à gauche)	GS
030	(Curseur vers le haut)	RS
031	(Curseur vers le bas)	US
032	(Espace)	

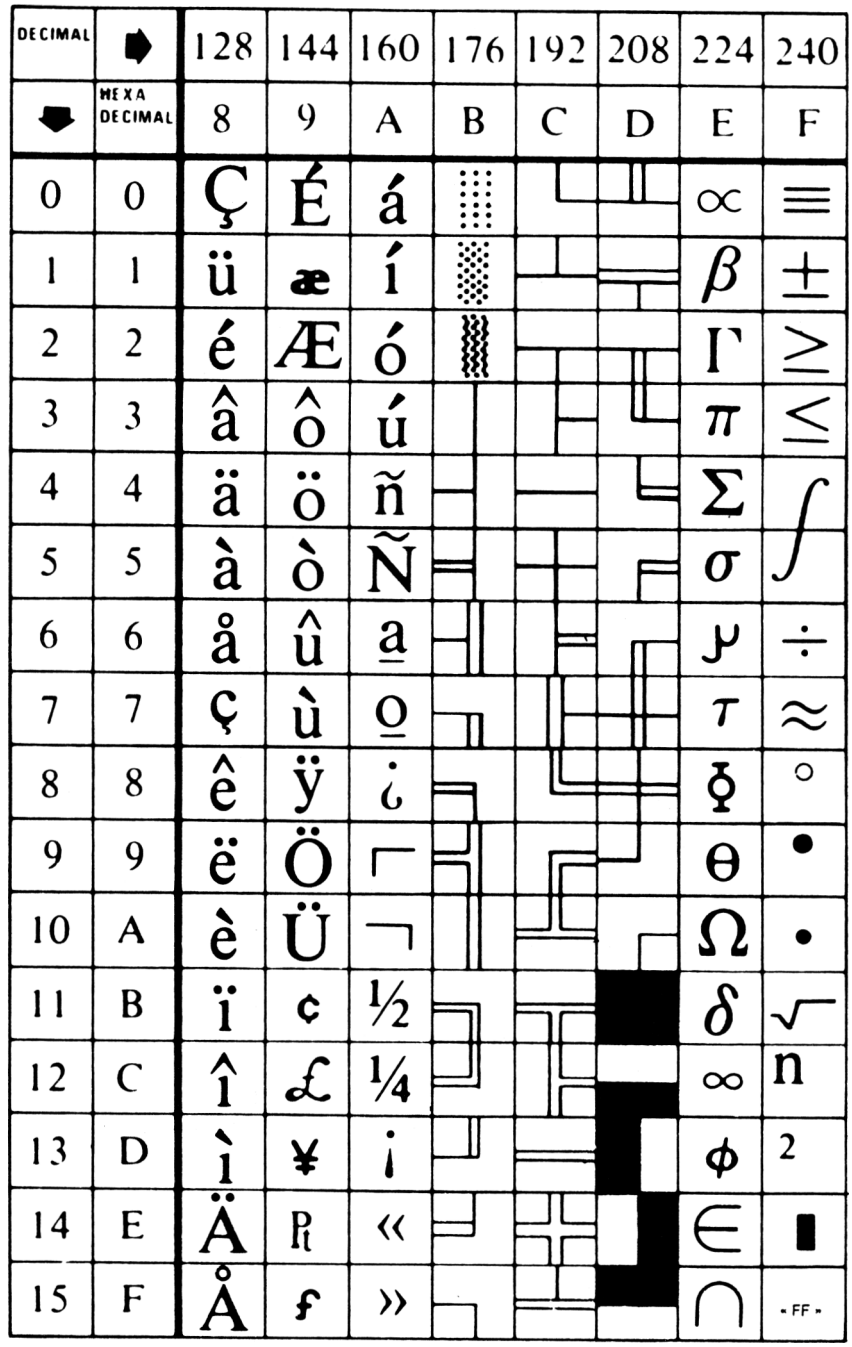
033	!	067	C
034	''	068	D
035	#	069	E
036	$	070	F
037	%	071	G
038	&	072	H
039	'	073	I
040	(	074	J
041	)	075	K
042	*	076	L
043	+	077	M
044	,	078	N
045	-	079	O
046	.	080	P
047	/	081	Q
048	0	082	R
049	1	083	S
050	2	084	T
051	3	085	U
052	4	086	V
053	5	087	W
054	6	088	X
055	7	089	Y
056	8	090	Z
057	9	091	[
058	:	092	\
059	;	093	]
060	<	094	^
061	=	095	_
062	>	096	'
063	?	097	a
064	@	098	b
065	A	099	c
066	B	100	d

Le code ASCII (suite)

101	e	153	Ö	205	═
102	f	154	Ü	206	╬
103	g	155	¢	207	╧
104	h	156	£	208	╨
105	i	157	¥	209	╤
106	j	158	Pt	210	╥
107	k	159	ƒ	211	╙
108	l	160	á	212	╘
109	m	161	í	213	╒
110	n	162	ó	214	╓
111	o	163	ú	215	╫
112	p	164	ñ	216	╪
113	q	165	Ñ	217	┘
114	r	166	ª	218	┌
115	s	167	º	219	█
116	t	168	¿	220	▄
117	u	169	⌐	221	▌
118	v	170	¬	222	▐
119	w	171	1/2	223	▀
120	x	172	1/4	224	α
121	y	173	¡	225	ß
122	z	174	«	226	Γ
123	{	175	»	227	π
124	¦	176	░	228	Σ
125	}	177	▒	229	σ
126	~	178	▓	230	µ
127	⌂	179	│	231	τ
128	Ç	180	┤	232	Φ
129	ü	181	╡	233	Θ
130	é	182	╢	234	Ω
131	â	183	╖	235	δ
132	ä	184	╕	236	∞
133	à	185	╣	237	∅
134	å	186	║	238	∈
135	ç	187	╗	239	∩
136	ê	188	╝	240	≡
137	ë	189	╜	241	±
138	è	190	╛	242	≥
139	ï	191	┐	243	≤
140	î	192	└	244	⌠
141	ì	193	┴	245	⌡
142	Ä	194	┬	246	÷
143	Å	195	├	247	≈
144	É	196	─	248	°
145	æ	197	┼	249	∙
146	AE	198	╞	250	·
147	ô	199	╟	251	√
148	ö	200	╚	252	n
149	ò	201	╔	253	²
150	û	202	╩	254	■
151	ù	203	╦	255	(Blanc ''FF'')
152	ÿ	204	╠		

CODE ASCII ILLUSTRÉ

DECIMAL	HEXA DECIMAL	0	16	32	48	64	80	96	112
		0	1	2	3	4	5	6	7
0	0	ESPACE (NULL)	►	ESPACE	0	@	P	‘	p
1	1	☺	◄	!	1	A	Q	a	q
2	2	☻	↕	"	2	B	R	b	r
3	3	♥	!!	#	3	C	S	c	s
4	4	♦	¶	$	4	D	T	d	t
5	5	♣	§	%	5	E	U	e	u
6	6	♠	▬	&	6	F	V	f	v
7	7	•	↨	’	7	G	W	g	w
8	8	◘	↑	(	8	H	X	h	x
9	9	○	↓	)	9	I	Y	i	y
10	A	◙	→	*	:	J	Z	j	z
11	B	♂	←	+	;	K	[	k	{
12	C	♀	∟	,	<	L	\	l	¦
13	D	♪	↔	—	=	M	]	m	}
14	E	♫	▲	.	>	N	^	n	~
15	F	☼	▼	/	?	O	_	o	△


DECIMAL
HEXA DECIMAL
128 144 160 176 192 208 224 240
8 9 A B C D E F
0 0 Ç É á ∝ ≡
1 1 ü æ í β ±
2 2 é Æ ó Γ ≥
3 3 â ô ú π ≤
4 4 ä ö ñ Σ ⌠
5 5 à ò Ñ σ ⌡
6 6 å û ª µ ÷
7 7 ç ù º τ ≈
8 8 ê ÿ ¿ Φ °
9 9 ë Ö ⌐ Θ •
10 A è Ü ¬ Ω •
11 B ï ¢ ½ δ √
12 C î £ ¼ ∞ n
13 D ì ¥ ¡ φ 2
14 E Ä ₧ « ∈ ■
15 F Å ƒ » ∩ -FF-

CODE ANSI

Le code ANSI fournit des commandes étendues pour le clavier et l'écran. Il est disponible via le fichier ANSI.SYS du DOS, qu'il convient de mettre en service en introduisant dans le CONFIG.SYS une ligne DEVICE = ANSI.SYS.

Résidant alors en mémoire centrale, il pourra être appelé par d'autres commandes (PROMPT, par exemple), ou par vos programmes. Il regroupe à son tour un certain nombre de commandes agissant sur le curseur, l'écran et le clavier ; en voici la liste, en 3 tableaux :

1. CURSEUR ET EFFACEMENT

Nom	Commande	Fonction	Commentaires
PCU	Esc[rangée;colonne*h*	Position du curseur	Déplace le curseur à la position spécifiée (par défaut à 1 ; 1)
CVH	Esc[nombredelignes*a*	Déplacement du curseur vers le haut	Sans changer de colonne
CVB	Esc[nombredelignes*b*	Déplacement du curseur vers le bas	Sans changer de colonne
CVD	Esc[nombredecolonnes*c*	Déplacement du curseur vers la droite	Sans changer de ligne
CVG	Esc[nombredecolonnes*d*	Déplacement du curseur vers la gauche	Sans changer de ligne
PHV	Esc[rangée;colonne*f*	Position horizontale et verticale du curseur	Déplace le curseur (par défaut, à 1 ; 1)
DPC	Esc[6*n*	Demande la position du curseur	Le contrôleur renvoie le RDC
EPC	Esc[rangée;colonne*r*	Etat de la position du curseur	Position du curseur avec périphérique d'entrée standard
MPC	Esc[*s*	Mise en mémoire de la position du curseur	On la retrouve avec la fonction suivante PAC
RDC	Esc[*u*	Repositionnement du curseur	Voir la fonction DPC
EE	Esc[2*J*	Effacement de l'écran	Efface l'écran et place le curseur à sa position initiale
EL	Esc[*k*	Effacement dans une ligne	Efface les caractères du curseur jusqu'à la fin de la ligne

2. ECRAN

Nom	Commande	Fonction
SCG	Esc[paramètre;... ;paramètre*m*	Spécifie l'attribut avec « paramètre » : Ø = Aucun attribut (blanc sur noir) 1 = Gras (surbrillance) 4 = Soulignement (monochrome) 5 = Clignotement 7 = Vidéo inverse 8 = Caractères invisibles (non affichés) 3Ø = Premier plan noir 31 = Premier plan rouge 32 = Premier plan vert 33 = Premier plan jaune 34 = Premier plan bleu 35 = Premier plan magenta 36 = Premier plan cyan 37 = Premier plan blanc 4Ø = Arrière-plan noir 41 = Arrière-plan rouge 42 = Arrière-plan vert 43 = Arrière-plan jaune 44 = Arrière-plan bleu 45 = Arrière-plan magenta 46 = Arrière-plan cyan 47 = Arrière-plan blanc
RTA	Esc[= paramètre*h* Esc[= *h* Esc[= Ø*h* Esc[?7*h* (au choix)	Réglage du type d'affichage ou de la largeur, selon le paramètre, avec : Paramètre Ø = 40 × 25 monochrome 1 = 4Ø × 25 couleur 2 = 8Ø × 25 monochrome 3 = 8Ø × 25 couleur 4 = 32Ø × 2ØØ couleur 5 = 32Ø × 2ØØ monochrome 6 = 64Ø × 2ØØ monochrome 7 = Retour marge automatique à la fin d'une ligne sur la suivante
ARMA	Esc[= paramètre*i* Esc[= *i* Esc[= Ø*i* Esc[?7*i* (au choix)	Annulation du retour automatique à la marge. Les paramètres sont identiques à RTA, sauf le 7 qui annule le retour automatique à la marge.

3. CLAVIER

Commande	Fonction
Esc[#;#;....#*p* Esc[''chaîne''....;*p* Esc[#;''chaîne''....# ;# ;« chaîne;#*p* ou toute autre combinaison de chaînes ou de chiffres décimaux	Le 1er code ASCII de la séquence donne le code à redéfinir. Le reste détermine la séquence déclenchée par la nouvelle définition. Si le 1er code est Ø (NUL), le 2e fournit un code ASCII étendu.

MOTS-CLES

Avec renvoi à leur page d'appel

A

Alt, 3, 6
Alt 124, 48, 56
Alt 255, 45
ANSI, 13
ANSI.SYS, 13, 91
APPEND, 109
Arborescence des répertoires, 35
ASCII, 3
ASSIGN, 77
Astérisque, 94
ATTRIB, 46
ATTRIBUTS DES FICHIERS, 62
AUTOEXEC.BAT, 19, 21
AZERTY, 16

B

Barre verticale coupée, 4, 56
Barre verticale brisée, 51
Basic, 128
Batch, 19
BAT., 19

C

Cadres, 6
Caractères condensés, 125
Caractères de remplacement, 94
CD, 139
Chaînage de commandes, 48
Chaîner des commandes, 51
Change Directory, 139
CHKDSK, 72
Classement, 50
Cluster, 58, 80
Code ASCII, 3, 158
COMP, 42
CON, 96, 98, 132
CONCATENATION, 97
CONFIDENTIALITE, 44
Console, 132
Contrôle Z, 30, 96, 116
Contrôle + Verrouillage majuscules, 88
COPY, 63, 96, 98, 114
COPY CON, 96
Ctrl + P, 118
Ctrl + PrtSc, 118
Ctrl + Z, 133, 152
Ctrl+Alt+F1, 16
Ctrl+Alt+F2, 16

D

Date, 27
Debug, 11, 125
Décerner un nom à chaque disque, 69
DIR, 25
*DIR *.*, 153
DIR/P, 25
DIR/W, 26
DISKCOMP, 42
DISKCOPY, 63
Disque virtuel, 130

E

ECHO, 30
ECHO OFF, 30
Editeur de lignes, 105
Edition, 2
EDLIN, 96, 100, 105
EEMS, 137
EFFACER LE CONTENU D'UN SOUS-REPERTOIRE, 141
EMS, 137
Enhanced Expanded Memory Specification, 137
ESC [, 13, 91
Expanded Memory Specification, 137

F

F1, 1
F3, 1
FAT, 58
Filtre, 56
Fin de fichier, 116
FIND, 56
FORMAT, 69, 74

H

Heure, 28

I

Impression en caractères condensés, 120
Interdire l'effacement d'un fichier, 46

K

KEYBFR.COM, 16

L

LABEL, 70
LASTDRIVE, 34
LPT1, 118, 120

M

Menus, 29
MENU.BAT, 30
Mode, 120
MORE, 88
Multitâche, 122

N

Norton utilities, 17, 76, 80

P

Paramètre remplaçable, 112
PATH, 34, 140, 147
PAUSE, 31
Point d'interrogation, 94
PRINT, 122
PRN, 118
PROMPT, 13, 27, 91, 145

Q

QWERTY, 16

R

RAMDISK.SYS, 131
Rechercher une chaîne de caractères, 56
Recopie d'un écran, 117
Récupérer un fichier effacé, 79
Redater un fichier, 114
Redirection, 103, 119
Répertoire actif, 145
Répertoire principal, 59, 138
Répertoire racine, 59, 138
Répertoires de rechange, 147

S

Shift + PrtSc, 117
SORT, 50
Sous-répertoires, 139
Sous-sous répertoires, 139
Spooler, 124
SUBST, 32

T

Table d'attribution des fichiers, 58
Touches d'édition, 2
Touches des claviers, 157
Trait vertical coupé, 48
Traits, 6
TREE, 35
Tris directs et inversés, 53
TYPE, 88, 99, 119

V

VDISK.SYS, 131
VERIFY, 42
VERIFY ON, 42
Verrouiller en position numérique, 11
VOL, 70
Volatile, 131

X

XCOPY, 63
XTREE, 35

Z

ZFORMAT, 74

Divers

+R, 46
-R, 46
/B, 116
$E], 13
$e], 91
$PSG, 91, 145

TABLE DES MATIÈRES

Avant-propos V

Clavier

1. Economisez les frappes des commandes répétitives grâce aux touches d'éditions 1
2. Comment afficher les caractères ASCII ne figurant pas sur le clavier ? 3
3. Comment tracer des cadres et filets parfaits 6
4. Comment démarrer avec le pavé numérique du clavier 102 touches en mode numérique 10
5. Comment reprogrammer les touches de fonctions et les autres avec ANSI.SYS 13
6. Comment reprogrammer un clavier « supplémentaire » 16

Confort

7. Comment automatiser l'exécution de commandes avec un AUTOEXEC.BAT 19
8. Comment passer automatiquement au programme d'application dès la mise en service du micro 21
9. Comment lire un répertoire qui ne tient pas en un seul écran 25
10. Comment afficher la date et l'heure 27
11. Comment créer un menu pour rendre le micro encore plus convivial. 29
12. Substituez une simple lettre à vos longs chemins d'accès 32
13. Si l'on se perd dans les multiples sous-répertoires, comment en afficher la structure ? 35
14. Comment permettre des réponses dans un fichiers .BAT 37
15. Comment afficher la date et l'heure en haut de l'écran 40

Commandes

16. Pour vérifier automatiquement les copies sur disque 42
17. Une méthode simple pour assurer la confidentialité des fichiers 44
18. Comment interdire l'effacement d'un fichier 46
19. Comment chaîner des commandes ? 48
20. Comment ordonner des informations par ordre alphabétique, numérique, ou mixte ? 50
21. Comment trier les informations à afficher ou imprimer ? 56
22. Structure du DOS sur un disque : ce qu'est la FAT 58

Disques

23. Comment copier d'une disquette à l'autre sur un micro avec une seule unité à disquettes 63
24. Comment afficher la capacité encore disponible sur le disque dur, et rien que cela 68
25. Comment décerner un nom à un disque 69
26. L'exploration des fichiers d'un disque 72
27. Ne reformatez pas accidentellement votre disque dur 74
28. Comment remplacer une unité à disque par une autre 77
29. Comment récupérer un fichier effacé par accident 79

Ecran

30. Comment afficher un document par écrans-complets ? 88
31. Comment afficher en surbrillance 91

Fichiers

32. Le bon usage des caractères de remplacement * et ? du DOS 94
33. Comment créer un fichier en le frappant directement au clavier 96
34. Comment ajouter des lignes dans AUTOEXEC.BAT ou CONFIG.SYS (ou les créer) 98
35. Comment enregistrer dans un fichier spécifique un répertoire, ou toute autre information 103
36. Un court fichier à créer ou modifier ? Utilisez EDLIN..... 105
37. Comment établir des chemins de recherche alternatifs pour les fichiers de données 108
38. Comment introduire des paramètres à la demande en lançant des fichiers .BAT 111
39. Comment redater un fichier..... 114

Imprimante

40. Comment imprimer ce qui apparaît à l'écran..... 117
41. Comment passer à l'impression en caractères condensés..... 120
42. Continuez à travailler sur votre micro pendant que l'imprimante sort votre listage 122
43. Comment programmer l'imprimante sous Debug..... 125
44. Comment programmer l'imprimante en Basic..... 128

Mémoires

45. Pour travailler plus vite, créez un disque virtuel..... 130
46. Pas assez de mémoire ? Voici comment accroître la capacité de la mémoire centrale 136

Répertoires

47. Répertoires et chemins d'accès : à quoi ça sert et comment les organiser..... 138
48. Quelques trucs relatifs aux sous-répertoires..... 141
    1. Effacer le contenu d'un sous-répertoire 141
    2. Remonter de plusieurs niveaux..... 142
    3. Revenir au répertoire principal..... 143
    4. Passer à un sous-répertoire de même niveau 143
    5. Sauter allègrement d'une branche à l'autre..... 144
49. Comment afficher le nom du sous-répertoire actif avec les caractères d'appel du DOS 145
50. Comment spécifier des répertoires de rechange..... 147
51. Comment appeler un programme directement, où qu'il se trouve sur le disque..... 150
52. Comment afficher la seule liste des sous-répertoires..... 153

Annexes

Les noms des touches du clavier 156
Le code ASCII 157
Mots-clés..... 161
Code ANSI..... 165

50 Recettes MS-DOS
1re Edition
H. Lilen

EDITIONS RADIO

Service lecteurs

(à retourner à Éditions Radio, 189, rue Saint-Jacques, 75005 Paris)

Pour nous permettre de vous proposer des ouvrages toujours meilleurs, nous souhaiterions recevoir vos critiques, appréciations et suggestions sur le présent livre :

_____

_____

_____

_____

Quels sont les ouvrages (thème, sujet, niveau) que vous souhaiteriez voir publier par notre société ?

_____

_____

_____

_____

Nous vous remercions de votre confiance et de votre coopération.

Éditions Radio

Je désire recevoir gratuitement et sans engagement (mettre une croix dans la case) :

☐ Votre catalogue général (Electronique professionnelle et grand public, Informatique, Hi-Fi, Vidéo, Photo)

Nom : _____ Prénom : _____

Adresse : _____

_____

Secteur d'activité et fonction : _____

_____

CENTRES D'INTÉRÊTS

☐ Electronique professionnelle
☐ Electronique de loisirs
☐ Vidéo
☐ Hifi, CB...
☐ Micro-informatique professionnelle
☐ Micro-informatique de loisirs
☐ Autres : .....
.....

Voir au dos "Correspondance Auteurs"

50 Recettes MS-DOS
1re Edition
H. Lilen

EDITIONS RADIO

Service lecteurs

Correspondance auteurs

(à retourner à Éditions Radio, 189, rue Saint-Jacques, 75005 Paris)

Pour toute demande d'éclaircissements techniques relatifs à ce livre, formulez ci-dessous vos questions, avec le maximum de précisions :

Nom : _____ Prénom : _____

Adresse : _____

ÉDITIONS RADIO

Comment...

Economiser les frappes des commandes répétitives
Afficher les caractères ASCII ne figurant pas au clavier
Tracer des cadres et filets parfaits
Démarrer avec le pavé numérique en mode numérique
Reprogrammer les différentes touches
Reprogrammer un clavier supplémentaire
Automatiser l'exécution de commandes
Lancer un programme dès la mise en route du micro
Lire un repertoire qui ne tient pas en un seul écran
Afficher la date et l'heure
Créer un menu
Substituer une simple lettre à vos longs chemins d'accès
Afficher la structure des sous-répertoires
Permettre des réponses dans un fichier .BAT
Afficher la date et l'heure en haut de l'écran
Vérifier automatiquement les copies sur disque
Assurer la confidentialité des fichiers
Interdire l'effacement d'un fichier
Chaîner des commandes
Ordonner des informations
Trier les informations à afficher ou imprimer
Structure du DOS sur disque : ce qu'est la FAT
Copier des disquettes avec une seule unité de disquettes
Afficher la capacité disponible sur le disque dur
Décerner un nom à un disque
Exploration des fichiers d'un disque
Ne pas reformater accidentellement votre disque dur
Remplacer une unité à disque par une autre
Récupérer un fichier effacé par accident
Afficher un document par écrans complets
Afficher en surbrillance
Le bon usage des caractères de remplacement * et ?
Créer un fichier en le frappant directement au clavier
Ajouter des lignes dans AUTOEXEC.BAT ou CONFIG.SYS
Enregistrer un répertoire dans un fichier spécifique
Utiliser EDLIN pour créer ou modifier un fichier court
Etablir des chemins de recherche alternatifs
Introduire des paramètres à la demande avec des fichiers .BAT
Redater un fichier
Imprimer ce qui apparaît à l'écran
Passer à l'impression en caractères condensés
Travailler pendant que votre imprimante sort un listage
Programmer l'imprimante sous Debug
Programmer l'imprimante en Basic
Créer un disque virtuel, pour travailler plus vite
Accroître la capacité de la mémoire virtuelle
Organiser répertoires et chemins d'accès
Effacer le contenu d'un sous-répertoire
Remonter de plusieurs niveaux
Revenir au répertoire principal
Passer à un sous-répertoire de même niveau
Afficher le nom du sous-répertoire actif
Spécifier des répertoires de rechange
Appel direct d'un programme, où qu'il se trouve sur le disque
Afficher la seule liste des sous-répertoires

9 782709 110242
ISBN 2 7091 1024 5
Code 77

ÉDITIONS RADIO

F 150/88/1

50 RECETTES DOS

H. LILEN